HP OpenView
Service Information Portal

Deployment and Integration Guide

Version: 3.2

Windows®, HP-UX, and Solaris

Legal Notices

Contents

Contents

4. Connecting SIP to Your Management Products

5. Configuring Users and Roles

Contents

6. Customizing Portal Views

7. Developing Portal Content

Contents

9. Configuring Authentication

10. Integrating SIP Into Your Own Environment

Contents

Contents

Contents

G. Uninstalling Service Information Portal

Support

Please visit the HP OpenView web site at:

`http://www.managementsoftware.hp.com/`

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

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

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

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

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

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

 HP Passport

 Active contract

 Premium contract

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

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

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

`https://passport.hp.com/hpp2/newuser.do`

1   Resources Available While Learning and Using SIP

What This Manual Covers

This manual covers several areas of SIP deployment and integration. It will guide you through the following tasks:

1. Planning your SIP deployment.

2. Connecting SIP to your management products.

3. Creating SIP users and roles.

4. Customizing portal views, including designing a custom look and feel to your portals.

5. Providing content through SIP modules, including integrating your own applications and data.

6. Segmenting management data by customers.

7. Implementing a supplied or custom authentication provider.

8. Integrating SIP into a specific environment, such distributed, wireless, non-English, and especially large environments.

9. Performing routine administrative tasks.

Available also is a glossary, as well as a set of appendixes that include important information such as:

- How SIP uses XML to store configuration data.

- The different methods by which the configuration data can be refreshed.

- A table of default protocol and port settings to reference if you are installing multiple management products on the same machine.

- How to perform some configurations directly through the XML files.

- Procedures for uninstalling SIP.

Getting Additional Documentation

Both printable and online documentation is available. The table below
lists printable documents available to you after SIP is installed. All
document files are stored in product directories under the following
directory:

Windows: `%SIP_HOME%\htdocs\C\manuals\`
UNIX: `/opt/OV/SIP/htdocs/C/manuals/`

Table 1-1   **HP OpenView Service Information Portal Documentation**

Document Title	Filename
SIP Installation Guide	`SIP_Install_Guide.pdf`
SIP Deployment and Integration Guide	`SIP_Deployment_Integration.pdf`
NNM Integration with SIP	`NNM_Integration.pdf`
OVO and Service Navigator Integration with SIP	`OVO_OVSN_Integration.pdf`
OVIS Integration with SIP	`OVIS_Integration.pdf`
OV Reporter Integration with SIP	`OVR_Integration.pdf`
OV Service Desk Integration with SIP	`OVSD_Integration.pdf`
OV Performance Insight Integration with SIP	`OVPI_Integration.pdf`
OV Performance Manager with SIP	`OVPM_Integration.pdf`

Online help is available when you need instruction on the SIP user
interface and how to use it. The list of help topics available from the
[Help] button on the main portal page differs depending upon the editing
permissions—ViewAdmin, UserPreferences, or ReadOnly—assigned to
the current role.

Below are sample screens that point out the various online help options.

Figure 1-1   **Online Help Available from the Main Portal Page**
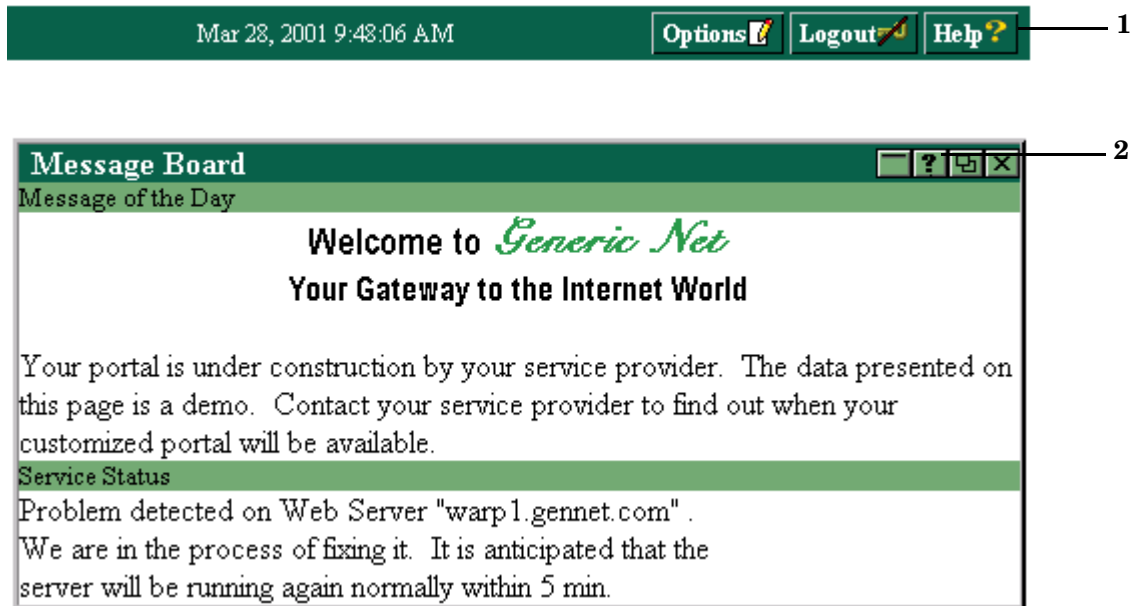
1 - The [Help] button displays help on performing tasks with the user interface. The list of displayed help topics differs depending upon the editing permissions granted to the user. The tasks described do not include module-specific information.

2 - The [?] button on a module title bar displays module help. These topics describe the module and data being displayed and are targeted to your end customers. A help topic for each module is provided by default with the SIP, but you can write your own help topic for a module and override the default topic.

Figure 1-2   **Online Help Available from the Options Page**
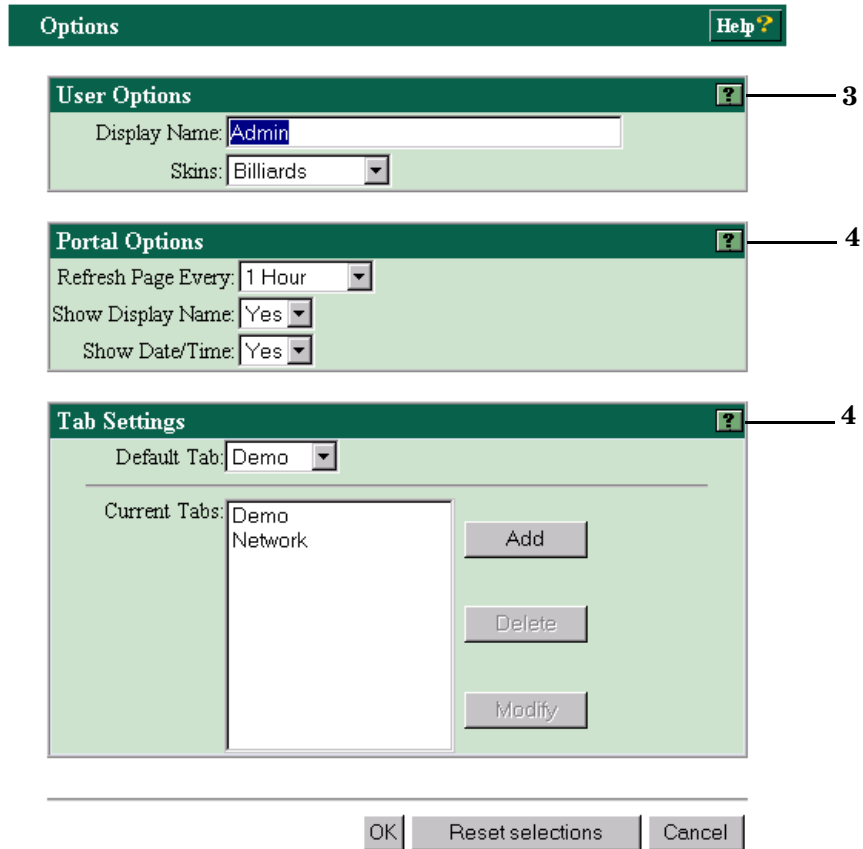
3 - The User Options section and [?] button are visible when the editing permissions level is set to UserPreferences or ViewAdmin for the current role. If a user has ReadOnly permissions, the [Options] button and page are not viewable.

4 - The Portal Options and Tab Settings sections and associated [?] buttons are only visible when the editing permissions level is set to ViewAdmin for the current role.

Figure 1-3   Online Help Available from Module Edit Pages
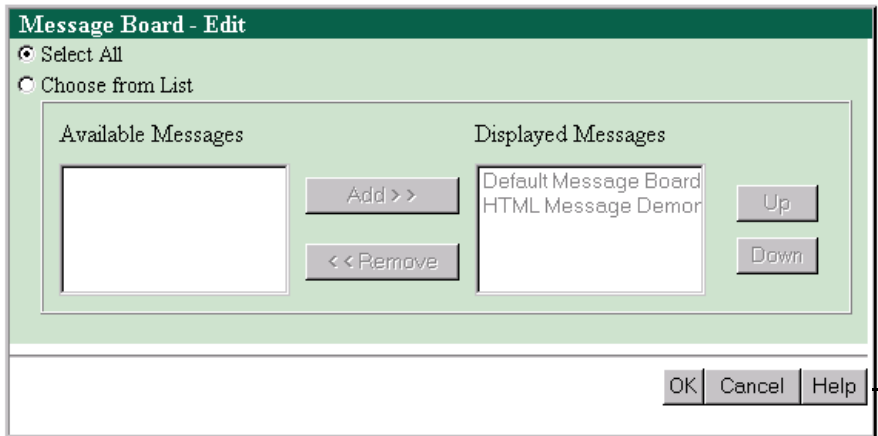
5 - The [Help] button on module Edit pages displays help on performing module editing tasks through the user interface.

2 Essential Concepts

Portal and Data Security

Service Information Portal (SIP) provides a secure environment for your managed data through several mechanisms.

Authorization

Authorization is the granting of access privileges to an authenticated user that determines what the user can see and do while logged into the system.

After a user is authenticated and considered a valid SIP user, authorization ensures that the user sees only the data you want him to see, and changes only the things you want him to change.

SIP uses an authorization model called the User Role model. Access rights are associated with roles. A **role** defines what a user can see and do through the portal at a particular point in time. By associating users with roles and assigning to each role what you want users in that role to be able to see and do, you achieve portal security.

Information in the Chapter 5, "Configuring Users and Roles," on page 75, will help you set up a secure user role configuration.

Filtering of Management Data

In SIP, management data is filtered at the role level. This involves explicitly associating a filter definition with each role. The filter defines the data to display when a user is acting in the given role. Filtering on a role-by-role basis provides data security, giving you a way to ensure that only relevant data gets displayed to a given user.

Information in the section "Defining/Modifying Management Data Filters" on page 90, will help you create filters for the user roles.

Customer Data Segmentation

Segmenting data by customer organization requires the use of a customer model: a mapping of customer organizations to their resources. This is yet another way to ensure that data gets displayed to the appropriate users.

Information in the Chapter 8, "Segmenting Data by Customer Organization," on page 215, will help you configure a fully-integrated customer model that contains customer-to-resource mappings.

Authentication

Authentication is the process by which a user identifies and validates himself/herself to the system.

SIP supplies several authentication providers from which you can choose, including ones that enable both outward- and inward-looking single sign-on (SSO). SIP also provides the functionality to integrate a custom authentication provider that you have created.

Information in Chapter 9, "Configuring Authentication," on page 245, will help you determine the type of authentication provider that best suits your needs, and explain exactly how to configure an authentication provider.

Firewalls and Proxy

Firewalls can be placed between the various tiers of the SIP distribution model for greater security. (See "Distribution Model" on page 24.)

SIP proxy capabilities allow you to (1) Provide restricted access to a web page, web data, web server, and so forth, and (2) Provide access through SIP to management product data without requiring the user to log in to the product. You can also provide the name and password that will be used when accessing the protected data.

Data Encryption Using Secure Socket Layer (SSL) Protocol

You can enable SIP to serve the portal using HTTPS, as well as enable SIP to be a client of HTTPS and get its back-end server content through SSL.

Information in "Setting Up SIP to Use Secure Socket Layer (SSL) Protocol for HTTPS" on page 274, will outline and guide you through the process of enabling SSL.

Distribution Model

SIP supports a three-tiered (Figure 2-1) as well as a four-tiered model (Figure 2-3) for both Windows and UNIX:

- Web Browser Tier
- Web Server Tier
- Application Server Tier
- Management Server Tier

Figure 2-1   **SIP Three-Tiered Distribution Model**
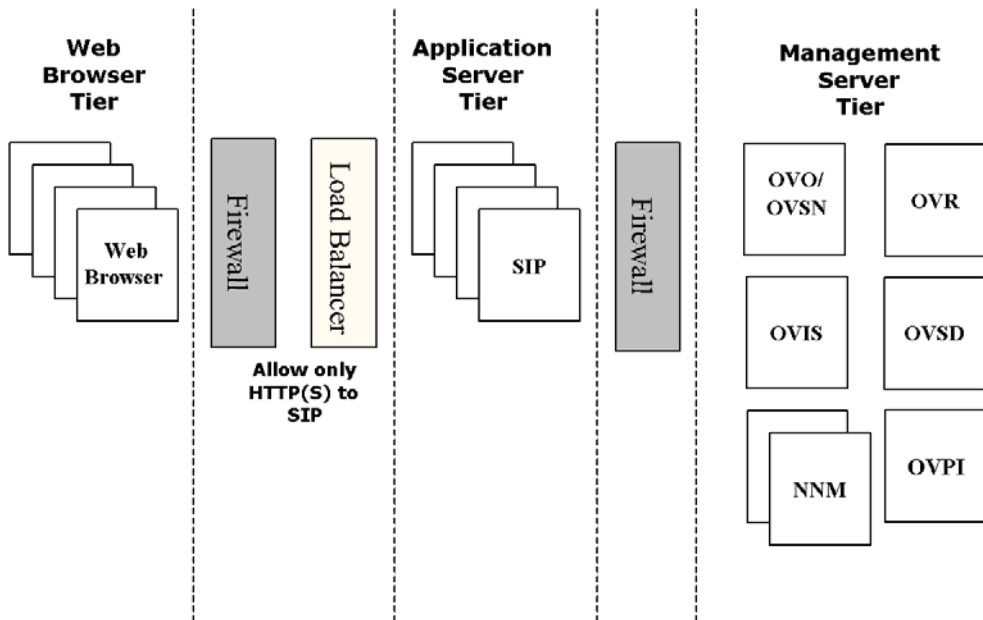
Figure 2-1 shows three tiers--the web browser tier, the application server tier, and the management server tier--separated by firewalls. Between the web browser tier and application server tier is also a load balancer.

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

The SIP server to management server communication can also go through a firewall. The ports that need to be opened through the firewall depend on the specific management products used. Figure 2-2 shows a distribution model for firewalls. For more detailed information, see "Configuring SIP in a Distributed Environment" on page 270.

Figure 2-2   Distribution Model (Firewalls)
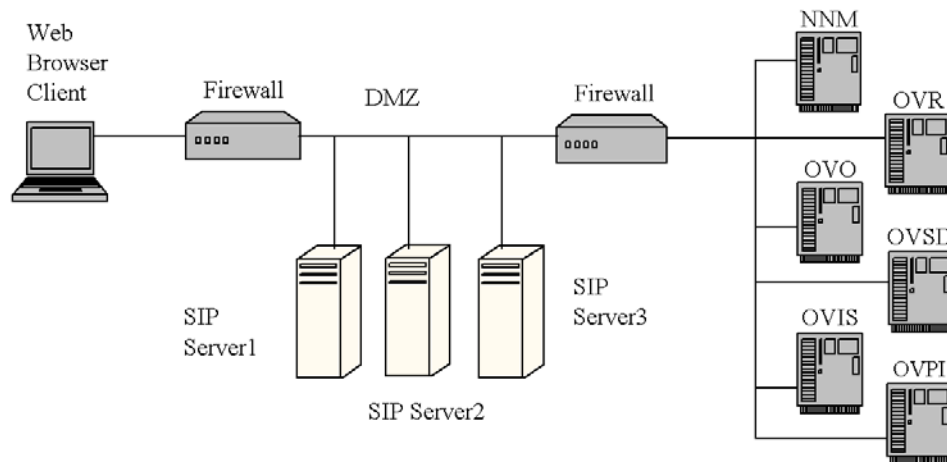
You can think of the overall architecture in terms of three parts: the extranet, the DMZ, and the intranet, with each being separated by a firewall. In Figure 2-2, the extranet is everything outside of the firewall and accessible to the outside world. The DMZ (demilitarized zone) is where the client software (SIP) is running. The intranet is where the data servers are running, such as NNM, OVO, and so forth.

Figure 2-3   **SIP Four-Tiered Distribution Model**
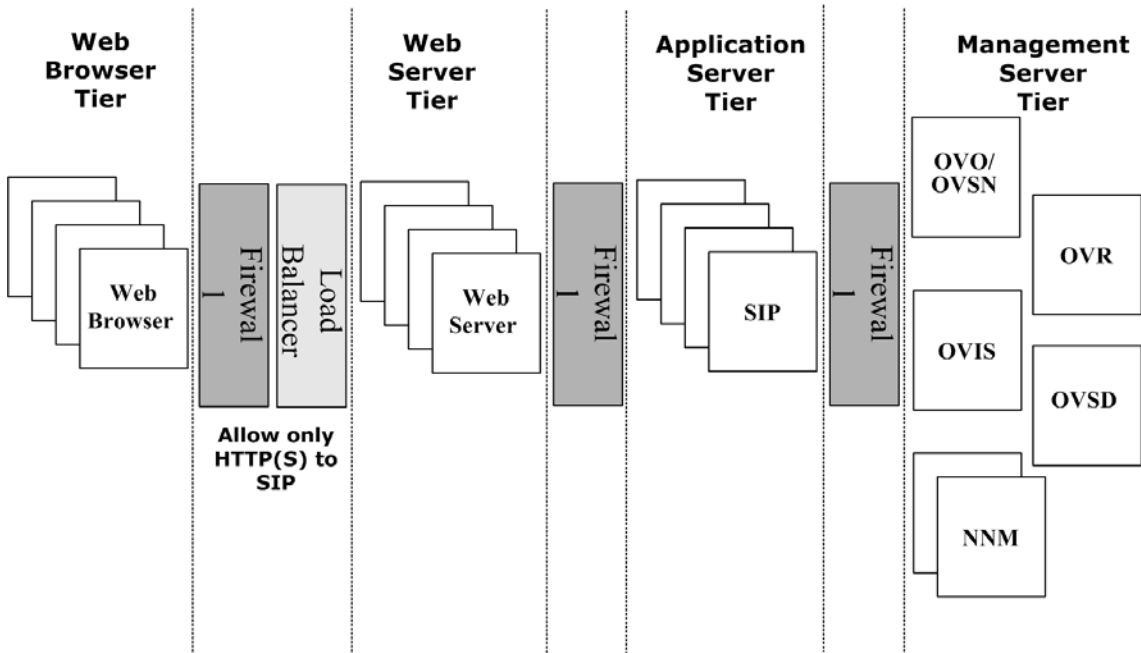
Figure 2-3 shows a four-tiered distribution model. The main difference between it and the three-tiered approach is that the web server is separated from the application server with a firewall in between. Communication between the web server and the application server occurs on a specific port using a specific protocol. By doing this, you can have the web server and the application server on separate machines.

Scalability Through Multiple Shared Servers

Scalability can be achieved through the use of multiple SIP servers, using a Web server load balancer that supports web session affinity. A shared SIP configuration can make it easier to manage the environment.

Figure 2-4   **Distribution Model (Shared Configuration)**
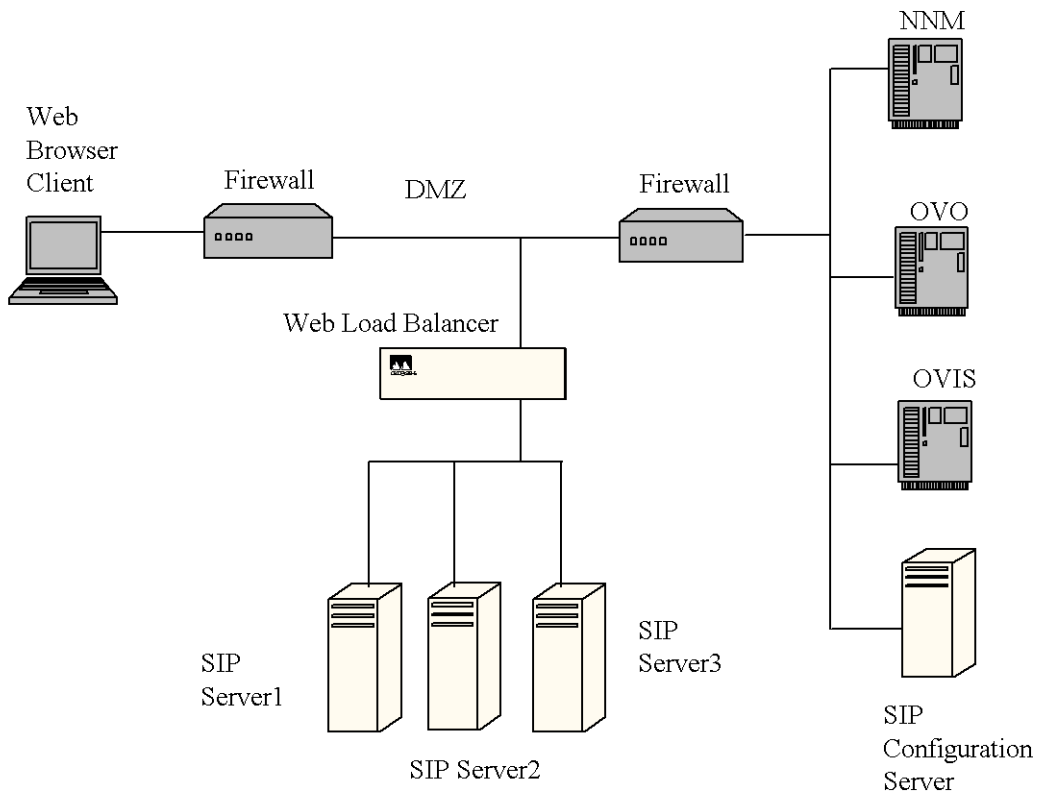
Figure 2-4 illustrates that you can share the SIP configuration across all SIP servers, allowing the SIP administrator to modify only one set of files that are shared globally, instead of having to manually keep the configurations of all three systems in sync, which is a cumbersome task with large installations.

I18N Model

Service Information Portal uses UTF-8 when communicating with the web browser and interpreting data in XML files. When getting data from management stations, SIP converts from the native codeset (such as Shift JIS) used by the management station to Unicode.

Figure 2-5 shows how the native codeset of the management data gets converted to Unicode and then displayed to the web browser in UTF-8 format.

Figure 2-5   **I18N Model**
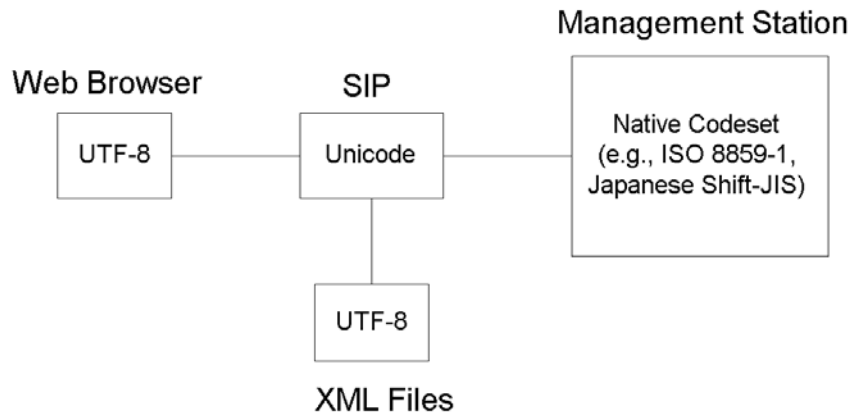
What You Can Expect From SIP

Service Information Portal:

- Installs under any language on any language system (such as Japanese versions of Windows).

- Uses locale-specific formats for date and time.

- Does locale-specific string comparison.

- Supports one Web browser codeset—UTF-8.

- Displays localized strings that vary based on the locale.

- Displays localized data from NNM (map data), Customer Views (customer names), OVIS, and OVSN.

- Supports non-ASCII string data specific to SIP as long as it is in the UTF-8 codeset. This includes:

  — Module titles specified in the module registration XML files and the portal view files

  — Tab names

  — User and role display names

  — Network Device Health gauge titles

  — Alarm category names

  — Generic module output data

  — Message Board message text

  — Bookmark link names

  — Customer names in the SIP customer model XML files

  — Help files

- Supports ASCII string data only for:

  — File and directory names

  — Login names

  — URLs

Limitations

- SIP does not ship the Java converters necessary to convert non-UTF-8 codeset data into UTF-8 codeset. SIP assumes that these will be part of the JRE on each platform.

- SIP has only been tested in a "mono-lingual" environment where all management servers were in the same locale.

NOTE   For more detailed information, see "Running SIP in Non-English Language Mode" on page 282.

3 Planning Your SIP Deployment

A Road Map to SIP Deployment and Integration

The following roadmap is a recommended approach to setting up, deploying, and integrating SIP. This is one of many possible approaches you could take. The iterative nature of some of the deployment tasks and the flexibility of the software sometimes require variations of this roadmap.

To provide the flexibility to use SIP in a way that makes the most sense for individual integrations, each step in the road map indicates whether the step is required or optional.

Table 3-1   Roadmap to SIP Deployment

Tasks	Required/ Optional	Manuals
1. Install and Verify SIP	Required	*SIP Installation Guide*
2. Migrate From the Previous Version	Optional	*SIP Installation Guide*
3. Plan Your SIP Deployment	Required	*SIP Integration and Deployment Guide*
4. Connect SIP to Your Management Products	Required	*SIP Integration and Deployment Guide* See the SIP manuals that document the integration of individual management products (see "Getting Additional Documentation" on page 17)
5. Configure Users and Roles	Required	*SIP Integration and Deployment Guide*
6. Customize Portal Views	Optional	*SIP Integration and Deployment Guide*

Table 3-1   **Roadmap to SIP Deployment (Continued)**

Tasks	Required/ Optional	Manuals
7. Develop Portal Content	Optional	*SIP Integration and Deployment Guide* To change the default module configurations, see the SIP manuals that document the integration of individual management products (see "Getting Additional Documentation" on page 17)
8. Segment Data By Customer Organization	Optional	*SIP Integration and Deployment Guide* Configuring the customer model sources varies depending upon the management product. See the SIP manuals that document the integration of individual management products (see "Getting Additional Documentation" on page 17)
9. Configure Authentication	Required	*SIP Integration and Deployment Guide*
10. Integrate SIP Into Distributed, Wireless, and Non-English Environments	Optional	*SIP Integration and Deployment Guide* Configuring individual modules for a non-English environment can vary. See the SIP manuals that document the integration of individual management products (see "Getting Additional Documentation" on page 17)

Making a Plan

The following questions and worksheets help you gather the information you need when creating customized portals and integrating SIP into your environment. The information gathering process covers all aspects of the roadmap. However, it does so in a slightly different order, as shown here:

- Portal Content

- Portal Users and Roles

- The Segmentation of Management Data By Customer

- Authentication

- The Integration of SIP into Distributed, Wireless, and Non-English Environments.

At the end of each section is a set of "Next Steps." After you complete the entire planning process, follow the steps to begin the deployment and integration of SIP.

Portal Content

The first step in the planning process is to decide what data you want to display through SIP:

- Content from other HP OpenView products through the use of HP OpenView integrations supplied with SIP.

- General content, such as links to bookmarks and messages displayed to your customers through a Message Board.

- Content from several third-party integrations (contributed and unsupported) that have been developed using SIP's Generic module.

- Content from your own applications and data.

Worksheets and Questions to Ask Yourself

In preparation for configuring SIP to display your content, it is recommended that you answer questions and fill in the worksheets in Table 3-2, Table 3-3, Table 3-4, and Table 3-5.

Go through the worksheets and indicate which modules you plan to use. The worksheets contain the following information:

- Description of the type of content you can display through SIP.

- Name of the SIP module that displays each type of content.

- Management product from which data comes.

Content from Supplied HP OpenView Integrations

Concerning Table 3-2, ask yourself these questions:

- What kind of information do you want to deliver/display to your customers?

- What HP OpenView management products do you want to integrate into SIP?

- Of the HP OpenView integrations supplied with SIP, which provide the kind of content you want to display to your users?

Table 3-2   **Content from Supplied HP OpenView Integrations**

Type of Content to Display through SIP	SIP Module	HP OpenView Management Product	Plan to Use? Yes/No
A collection of alarm messages gathered from one or more alarm categories within NNM.	Alarms	Network Node Manager	_____
A collection of submaps from one or more NNM maps. Each map must be open on the NNM management station before the desired submap can be displayed in SIP. You can drill-down through the NNM submap hierarchy.	Topology Map	Network Node Manager	_____
Custom gauges that track the health of network devices. Your customers can monitor network performance at-a-glance.	Network Device Health	Network Node Manager	_____
Information from the OVO message database: • A high-level list of the type and number of messages received within one or more message categories. • A complete list of messages received within a specific category. • All known information about a specific message	OVO Messages	HP OpenView Operations for UNIX HP OpenView Operations for Windows	_____

Table 3-2 **Content from Supplied HP OpenView Integrations (Continued)**

Type of Content to Display through SIP	SIP Module	HP OpenView Management Product	Plan to Use? Yes/No
A table-like view of services. You can drill-down to details, including actions, propagation and calculation rules, if configured within OVSN.	Service Browser	HP OpenView Service Navigator	_____
A business-card view of a service. You can drill-down to details, including actions, propagation and calculation rules, if configured within OVSN.	Service Cards	HP OpenView Service Navigator	_____
Hierarchical view of the OVSN configuration settings for a particular service.	Service Graph	HP OpenView Service Navigator	_____
A gauge-like view of aggregated service status. You can drill-down to details, including actions, propagation and calculation rules, if configured within OVSN.	Service Health	HP OpenView Service Navigator	_____
Your own custom presentation of Service Navigator data.	Custom Service	HP OpenView Service Navigator	_____
A set of gauges, charts, and graphs showing information collected by OVIS. Information is collected through the monitoring of standard internet services such as HTTP, HTTPS, DNS, SMTP, and POP3.	Internet Services	HP OpenView Internet Services	_____

Table 3-2   **Content from Supplied HP OpenView Integrations (Continued)**

Type of Content to Display through SIP	SIP Module	HP OpenView Management Product	Plan to Use? Yes/No
The ability for Service Desk end users to submit, view, or modify service calls.	Service Desk	HP OpenView Service Desk	_____
The ability for Service Desk specialists to view or modify incidents, problems, changes, and work orders. Specialists can also submit, view, or modify service calls.	Service Desk for Specialists	HP OpenView Service Desk	_____
Reports generated by HP OpenView Reporter after gathering information from HP OpenView Operations and HP OpenView Service Navigator.	OVO Reports	HP OpenView Reporter	_____
Reports generated by HP OpenView Reporter after gathering information from NNM.	NNM Reports	HP OpenView Reporter	_____
Reports generated by HP OpenView Reporter from HP OpenView Performance.	OV Performance Reports	HP OpenView Reporter	_____
Reports generated by HP OpenView Performance Insight that are configured and deployed to your Performance Insight Web Access Server.	Performance Insight and Performance Insight Browser modules	HP OpenView Performance Insight	_____

Table 3-2   Content from Supplied HP OpenView Integrations (Continued)

Type of Content to Display through SIP	SIP Module	HP OpenView Management Product	Plan to Use? Yes/No
A set of tables, graphics, and gauges showing information collected by OVPM. Information is collected by monitoring system resource utilization and performance.	OVPM Module	HP OpenView Performance Manager	_____

General Content

Concerning Table 3-3, ask yourself these questions:

- Do you want to provide your customers with links to useful web sites through the Bookmarks module?

- Do you want to take advantage of the SIP Message Board to communicate with your customers?

Table 3-3   General Content

Type of Content to Display through SIP	Module to Use	Plan to Use? Yes/No
Links to URLs	Bookmarks URLs: _____ _____ _____ _____ _____ _____ _____	_____
Messages from you to your customers (announcements, message of the day, and so forth.)	Message Board Messages: _____ _____ _____ _____ _____	_____

Content from Contributed (Unsupported) Integrations

Concerning Table 3-4, ask yourself this question:

- Of the unsupported contributed modules provided with SIP, which provide the kind of content you want to display to your users? (Be aware that some of these modules have limitations, such as not supporting proxy or customer segmentation.)

Table 3-4   **Content from Contributed (Unsupported) Integrations**

Type of Content to Display through SIP	Integration Modules	Plan to Use Yes/No?
A view of the Concord Net Health Web UI.	Concord	_____
A view of the Crystal Enterprise Web UI.	Crystal	_____
A view of the Keynote web site.	Keynote	_____
An NNM 6.2 view of the network discovered using the Cisco Discovery Protocol.	NNM_CDP	_____
A view of NNM 6.2 Command Results.	NNM_Commands	_____
A view of NNM 6.2 Node Details.	NNM_NodeDetails	_____
A view of NNM 6.2 Node View.	NNM_NodeView	_____
A view of NNM 6.31 Node View.	NNM6_31NodeView	_____
A view of NNM reports proxied by the SIP server so the browser does not need direct access to the NNM server.	NNM_Reports	_____
A view of NNM 6.2 Show Path	NNM_ShowPath	_____
A view NNM Web Applications.	NNM_Web	_____
A view of the Opticom iView reports.	Opticom	_____
A view of HP OpenView Problem Diagnosis 1.0.	ProblemDiagnosis	_____
A view of the Remedy ARWeb 4.x interface proxied through the Apache web server.	Remedy	_____

Table 3-4   **Content from Contributed (Unsupported) Integrations**

Type of Content to Display through SIP	Integration Modules	Plan to Use Yes/No?
The ability for Service Desk 3 end users to view and modify current service tickets. End users can also drill-down for detailed information on a specific ticket and submit a new ticket.	ServiceDesk3	_____
A view of VantagePoint Reporter Reports.	VP_Reporter	_____
A view of WebTrends Reports.	WebTrends	_____
A view of Yahoo Headlines. Users who already have a myYahoo account, will see their personalized view.	Yahoo_Headlines	_____

Content That Integrates Your Own Applications and Data

Concerning Table 3-5, ask yourself these questions:

- What additional content do you want to provide that is not supplied with SIP:

  — Display content from a URL? For example, if you have an HTML file on a different server or a CGI program or other web application, you can display the contents of the URL.

  — Display output from executable commands? For example, if you want to run a command, such as `ping` or a command that generates results from a database, and display the output of the command in the portal.

  — Display HTML from an external file? For example, if you have a report in an HTML file on the local machine or a tool that generates reports in the form of text, you can display those reports through the portal.

  — Transform XML to a displayable format using XSLT?

  — Display embedded HTML? For example, if you have a tool that generates reports in the form of a GIF image, you can display the report through SIP.

Table 3-5   Content That Integrates Your Own Applications and Data

Integrations to be Developed	New Module Entries
New Module: _____	_____
New Module: _____	_____
New Module: _____	_____

Next Steps

Continue making your plan. When you are finished:

- Display content from supplied HP OpenView integrations by connecting SIP to your management products (Road Map, Task 4). You can then configure the default modules, if desired (Road Map, Task 7).

- Display general content through SIP by creating bookmarks and Message Board messages (Road Map, Task 7, Develop Portal Content).

- Display content from contributed (unsupported) integrations by registering the contributed modules with SIP (Road Map, Task 7, Develop Portal Content).

- Display content that integrates your own applications and data by developing your own modules (Road Map, Task 7, Develop Portal Content).

Portal Users and Roles

The second step in the planning process is to identify to whom you want to display data, and understand how to grant them authorization to SIP.

SIP uses an authorization model called the User Role model. Access rights are associated with roles. A **role** defines what a user can see and do through the portal at a particular point in time. By associating users with roles and assigning to each role what you want users in that role to be able to see and do, you achieve portal security.

A basic role consists of four parts, three of which are required:

- **A portal view** (required)

  A portal view is what is displayed to a user. It is a configured set of modules and how they appear on tabs. It is also a configured set of portal view attributes, such as name in the button bar, portal skin, refresh rate, default tab, portal header, and portal footer, and so forth.

  Several portal views are supplied as samples that you can use as starting points. Be aware that any changes you make to an original portal view will not be made automatically in the portal views that are based upon the original.

- **An editing permission level** (required)

  Editing permission levels define the interactive editing operations that a user can perform through the portal interface. Each level of editing permissions includes all the operations defined by the previous level and some additional operations. The three levels are: ReadOnly, UserPreferences, and ViewAdmin.

- **A management data filter** (required)

  Management data refers to information about resources, such as services, nodes, interfaces, servers, and so forth, that you want to display through the portal.

  In SIP, you must filter management data at the role level, explicitly associating a filter definition with each role. Filtering on a role-by-role basis provides data security. This gives you a way to ensure that only relevant data gets displayed to a given user.

  You have three basic alternatives for filtering management data:

— Use the `AllData` filter to allow users of the associated role to see all data from your management stations.

— Use the `NoData` filter to allow users of the associated role to see no data from your management stations.

— Use a management data filter that specifies one or more organizations in the SIP customer model. The SIP customer model is a mapping of organizations to their resources (nodes, interfaces, and services). Only the data associated with a specified organization is displayed, thus, segmentation of customer data can be accomplished with this third filtering alternative.

• **An extensible list of role properties** (optional)

Use role properties to do the following:

— Display customer-segmented data from a management product that has its own customer model that is independent of SIP's Simple Customer Model.

— Implement a single sign-on solution for logging in to back-end management applications. This solution can be used by modules, including the generic module.

Before You Begin Planning

To help you better understand the User Role model and how it works, look at Figure 3-1 on page 48. It shows the sample users, roles, and views that are supplied with SIP. From this diagram, you can see that:

• Users can be assigned to multiple roles.

• A role defines a portal view, edit permissions level, management data, and optional role properties.

• Portal views are essentially a set of tabs and modules.

After looking at the diagram, log into SIP as one of the users listed in the left-hand column of the diagram. In the SIP interface, switch among the roles and tabs, and correlate what you see in the SIP interface with the configuration in the diagram. When you are finished looking at the SIP interface, go to "Questions to Ask Yourself" on page 49.

Starting SIP

- Open a browser window and enter the URL that starts SIP:

http://*yourhostname*/ovportal

NOTE *Windows only*: You can also start SIP by selecting:
`Start:Programs->HP OpenView->Service Information`
`Portal->Service Information Portal`

NOTE *UNIX only*: If you configured the SIP web server to a port other than
80, use the following URL instead, where port is the configured web
server port for SIP: **http://*yourhostname:port*/ovportal**

If you are going to run SIP with a secure web server, the URL for
accessing SIP is: **https://*yourhostname:port*/ovportal**

NOTE Until you connect SIP to your management products, the modules will
display the message "Currently Not Configured."

Figure 3-1 shows the sample users, roles, and views supplied with SIP.
Each supplied user is associated with one or more roles; and each role is
associated with one and only one portal view, management data filter,
and edit permissions level.

Figure 3-1   Sample User/Role/View Configurations Supplied With SIP
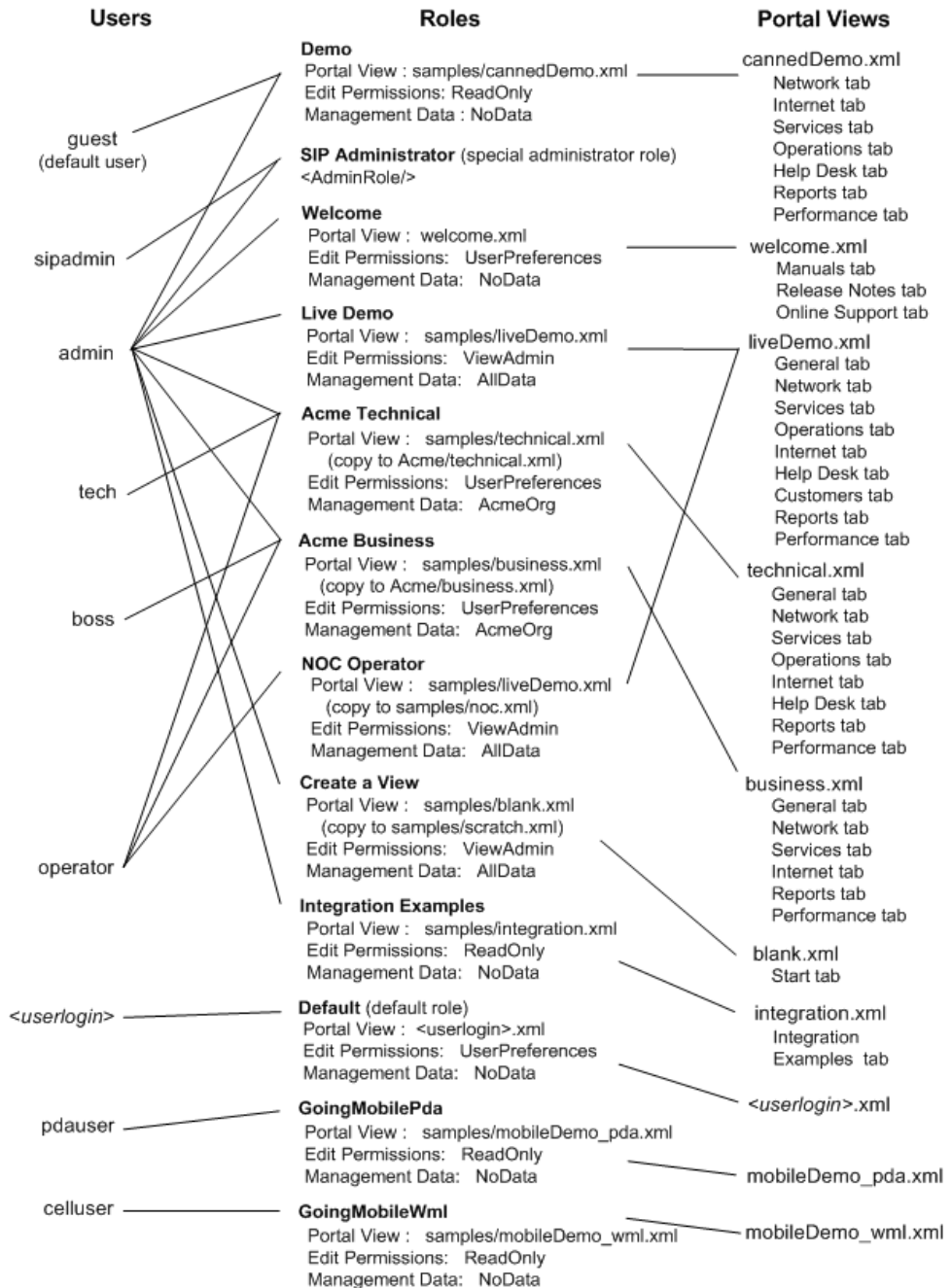
Questions to Ask Yourself

- Who will be using the portal? Customers? Organizations within a company? Business people, technical people, NOC users?

- Can you categorize the content you want to display based on groups of users, such as business users, technical users, NOC users, and so forth? Do you want to display some modules to business users but not to technical users? For example, business users may be more interested in report modules. Technical users may be more interested in modules that display the details of network and service health.

- Can you identify distinct roles for your users? For example, Operator role, CustomerABC administrator role, technical user role, business user role, and so forth.

- Do you want to give limited administrative privileges to certain customer logins so they can add and configure modules in others' portal views?

- Do you want to use a default role for some or all of your users? If so, do not explicitly configure a user entry for each login. Instead, configure a portal view file specifically for the login.

- Do you want to make available a "default user" who has access to a limited portal? With this user, you might, for example, allow visitors to view SIP without letting them see real management data or be able to perform editing operations.

- Do you need maximum flexibility in defining specifically what each user can see and do? If so, you are required to configure a user and initial role in the User Role model for each valid login?

- Is there a logical grouping of users and roles that you want to manage as one entity? For example, you can create a user role package that contains all of the users, roles, and management data filters associated with one customer organization.

- What modules do you want to display to each identified role? What will you name the tabs that the modules appear on and how will you organize the tabs?

- Can you leverage the supplied portal views as starting points for your own portal views? (See Figure 3-1 on page 48.)

- Do you want to customize the portal look and feel by using your own graphic images in the portal banner and footer?

- Do you want to tailor the portal look and feel to individual customers?

- Are the supplied portal skins (color schemes and fonts) sufficient for your portals, or do you want to create custom portal skins? (Portal skins are changed through the [Options] on the SIP button bar.

Worksheets

The following worksheet (Table 3-6 on page 51) is an empty portal view configuration. Following it (Table 3-7 on page 52) is an empty user role configuration for one customer with three users. When filling out Table 3-7 to create a user role configuration for one of your customer, use Table 3-8 on page 54 as an example.

NOTE   Until you complete the task of creating a customer model (Road Map, Task 8), you cannot define the MgmtData and Properties for a role. In the meantime, however, you can use one of the default management data filters AllData or NoData, and leave the Properties blank.

Table 3-6   Blank Worksheet: Portal View

Portal View	Tabs, Columns, Modules	Attributes
Portal View Directory and File Name (*<directory>*/*<portalview>*.xml): _____	Tab: _____ Column 1 width: narrow or wide? _____ Module: _____ Module: _____ Module: _____ Column 2 width: narrow or wide? _____ Module: _____ Module: _____ Module: _____ Tab: _____ Column 1 width: narrow or wide? _____ Module: _____ Module: _____ Module: _____ Column 2 width: narrow or wide? _____ Module: _____ Module: _____ Module: _____	Default Tab: _____ Refresh Rate: _____ Skin: _____ Show Date and Time? Y/N _____ Show User Name? Y/N _____ User Name? _____ Help Topic: custom or default? _____ Header: custom or default? _____ Footer: custom or default? _____

Table 3-7   **Blank Worksheet: User Role Configuration for Customer:**_____

Users	Roles Associated with User	Portal View Associated with Role
User: _____	Role: _____ Portal View: _____ copy on write:_____ Edit Permissions: _____ Mgmt Data: _____ Properties: name=_____ value=_____ name=_____ value=_____ name=_____ value=_____ name=_____ value=_____ Role: _____ Portal View: _____ copy on write:_____ Edit Permissions: _____ Mgmt Data: _____ Properties: name=_____ value=_____ name=_____ value=_____ name=_____ value=_____ name=_____ value=_____ Role: _____ Role: _____	Portal View: _____.xml Tab:_____ Modules: _____ _____ _____ Tab:_____ Modules: _____ _____ _____ Portal View: _____.xml Tab:_____ Modules: _____ _____ _____ Tab:_____ Modules: _____ _____ _____

Table 3-7   **Blank Worksheet: User Role Configuration for
Customer:**_____

Users	Roles Associated with User	Portal View Associated with Role
User: _____	Role: _____ Portal View: _____ copy on write:_____ Edit Permissions: _____ Mgmt Data: _____ Properties: name=_____ value=_____ name=_____ value=_____ name=_____ value=_____ name=_____ value=_____	Portal View: _____.xml Tab:_____ Modules: _____ _____ _____ Tab:_____ Modules: _____ _____ _____ Tab:_____ Modules:_____ _____ _____
User: _____	Role: _____ Portal View: _____ copy on write:_____ Edit Permissions: _____ Mgmt Data: _____ Properties: name=_____ value=_____ name=_____ value=_____ name=_____ value=_____ name=_____ value=_____	Portal View: _____.xml Tab:_____ Modules: _____ _____ _____ Tab:_____ Modules: _____ _____ _____ Tab:_____ Modules:_____ _____ _____

Table 3-8   **Sample User Role Configuration for Customer: BestCorp**

Users	Roles Associated with User	Portal View Associated with Role
adminBestCorp	**BestCorp Admin**	**customeradmin.xml**
	Portal View: samples/customeradmin.xml copy on write: BestCorp/customeradmin.xml	General tab: Bookmarks and Message Board modules
	Edit Perms: ViewAdmin	Customers tab: User Role Information and Managed Resources modules
	Mgmt Data: BestCorpOrg	
	Properties: VPIS.server=aa.bbb.com VPIS.customer=BC VPIS.password=bcuser	
	Properties: OVPM.server=aa.bbb.com OVPM.customer=BC OVPM.password=bcuser	-------------------------------------
	ServiceDesk.SSPserver=xx.yyy.com ServiceDesk.userName=BC ServiceDesk.password=bcSD ServiceDesk.virtualDirectory=sd-sp4	**blank.xml** Start tab: no modules
	OVPI.server=jj.kkk.com OVPI.userName=BC OVPI.password=bcpi	
	---	-------------------------------------
	Create a View	
	Portal View: samples/blank.xml copy on write: BestCorp/scratch.xml	-------------------------------------
	Edit Perms: ViewAdmin	
	Mgmt Data: BestCorpOrg	

	BestCorp Technical	

54	**BestCorp Business**	**Chapter 3**

Table 3-8   **Sample User Role Configuration for Customer: BestCorp**

Users	Roles Associated with User	Portal View Associated with Role
tech	**BestCorp Technical**	**technical.xml**
	Portal View: samples/technical.xml copy on write: BestCorp/technical.xml	General tab: Bookmarks and Message Board, Managed Resources modules
	Edit Perms: UserPreferences	Network tab: Network Device Health, Alarms, Topology modules
	Mgmt Data: BestCorpIT	
	Properties: VPIS.server=aa.bbb.com VPIS.customer=BC VPIS.password=bcuser	Services tab: Service Cards, Service Health, Service Browser, Service Graph modules
	ServiceDesk.SSPserver=xx.yyy.com ServiceDesk.userName=BC ServiceDesk.password=bcSD ServiceDesk.virtualDirectory=sd-sp4	Operations tab: OVO Message modules
		Internet tab: Internet Services modules
	OVPI.server=jj.kkk.com OVPI.userName=BC OVPI.password=bcpi	Help Desk tab: Service Desk for Specialists module
		Reports tab: OV Performance Reports module
		Performance tab: OV Performance Insight Browser module

Table 3-8   **Sample User Role Configuration for Customer: BestCorp**

Users	Roles Associated with User	Portal View Associated with Role
boss	**BestCorpBusiness**	**business.xml**
	Portal View: samples/business.xml copy to: BestCorp/business.xml	General tab: Bookmarks and Message Board modules
	Edit Perms: UserPreferences	Network tab: Network Device Health, Topology modules
	Mgmt Data: BestCorpMgmt	Services tab: Service Cards, Service Health, Service Browser, Service Graph modules
	Properties: VPIS.server=aa.bbb.com VPIS.customer=BC VPIS.password=bcuser	Internet tab: Internet Services modules
	ServiceDesk.SSPserver=xx.yyy.com ServiceDesk.userName=BC ServiceDesk.password=bcSD ServiceDesk.virtualDirectory=sd-sp4	Reports tab: OV Performance Reports module
	OVPI.server=jj.kkk.com OVPI.userName=BC OVPI.password=bcpi	Performance tab: OV Performance Insight module

Next Steps

Continue making your plan. When you are finished:

- Configure the User Role model by creating users and roles through the SIP Configuration Editor (Road Map, Task 5).

- Create a custom portal look and feel by customizing the portal views (Road Map, Task 6).

The Segmentation of Management Data By Customer

The third step in the planning process is to think about your customers and how to display management data that is relevant to them. This involves providing SIP with a mapping between your customers and their resources.

Segmenting data by customer organization requires the use of a customer model: a mapping of customer organizations to their resources. Table 3-9 shows which supplied modules use the SIP customer model, which use their own customer model, and which do no data filtering.

Table 3-9   **The Use of Customer Models By Supplied Modules**

Management Software/ Modules	Use the SIP Customer Model	Use Their Own Customer Model	Do No Filtering of Management Data
NNM: Alarms, Topology Map, and Network Device Health modules	X		
OVIS: Internet Services module		X	
OVPM: Performance Manager Module		X	
OV Reporter modules	X		
OVO/OVSN modules	X		
OVSD modules		X	
OVPI modules		X	
Message Board module			X
Bookmarks module			X

Questions to Ask Yourself

Answers to the following questions will help you prepare to display, through SIP, management data that is segmented by customer organization.

- If you use Customer Views for NNM, do you want to use the customer model configured through Customer Views to display segmented NNM data in SIP?

- If you do not use Customer Views for NNM, do you still want to display segmented NNM data in SIP? You can create a customer model source that retrieves resource data from the NNM object database, and create a customer model source that maps the resources to organizations.

- Besides NNM data, is there other data that you want to segment by nodes or interfaces?

- Do you want to display customer-segmented service information from HP OpenView Service Navigator? If so, you will need to set up a customer model source (an XML file) that contains mappings from customers to services.

- Are you displaying data from products that have their own customer models, such as HP OpenView Internet Services and HP OpenView Service Desk?

- Do you want to display customer-segmented NNM Reports? Do you want to display OV Performance Reports? Both are generated by OV Reporter by exporting the SIP customer model to those products.

- Do you have customer model data in another product or database that you want to import to SIP? If so, you will need to create a custom customer model data source.

- Do you have another product or database to which you want to export the SIP customer model? (This would allow the other product to segment data based on the SIP customer model.)

Next Steps

Continue making your plan. When you are finished:

- Set up the SIP customer model by developing the customer model sources and registering them with SIP (Road Map, Task 8). This task varies depending upon the individual management product, so see the SIP manuals that document the individual management products.

- Export the SIP customer model to NNM or OV Reporter through supplied CGI programs and servlets. Register the export destinations with SIP (Road Map, Task 8).

Authentication

The fourth step in the planning process is to consider the mechanism by which SIP users will be authenticated. Authentication is the process by which users identify and validate themselves to the system.

SIP supplies several authentication providers to choose from. If none of the supplied authentication providers meets your needs, you can write and integrate your own custom authentication provider. SIP provides APIs and sample code to write your own Java class to integrate with SIP.

Of the supplied authentication providers, there are two types: those that do portal authentication and those that do external authentication.

Portal Authentication

The portal itself can display a login page and perform authentication using services provided by the configured authentication provider. In this model, the user logs in to the portal, and can log out from the portal independently of login to any other authentication system.

In this type of authentication model, SIP gathers the user name and password from the user and then uses the authentication services or data source defined for the authentication provider to determine if the user is authenticated.

External Authentication

In this type of authentication provider, SIP never sees the password for the user. Some authentication service determines that the user has been authenticated and merely provides SIP with the user name. Based on the configured external authentication provider, SIP trusts that the user has been appropriately authenticated. The scope and duration of the user's authentication are defined externally, and are not a concern of SIP. SIP need only recognize whether or not the user is authenticated.

An external authentication provider can be used to make SIP a part of a larger single sign-on solution.

The Web Server Authentication Provider provided with SIP is an example of an external authentication provider. If the web server supports one of the various forms of authentication performed by the

Web Server (HTTP Basic, HTTP Digest, HTTPS Client, or Windows NT Challenge/Response), and the portal administrator has configured it, the authentication provider class must merely recognize whether a user has already been authenticated. In this model, the scope and duration of the user's authentication are defined by the Web Server.

Questions to Ask Yourself

It is recommended that you answer the following questions for yourself and then fill out the worksheet in preparation for configuring an authentication provider. For information on portal authentication versus external authentication, see

- Do you want SIP to perform the authentication or do you want an external mechanism to authenticate the user? The external mechanism can communicate to SIP whether the user has been authenticated.

- Will your portal users log in to SIP and log out of it as a standalone program, or will they run SIP from some other program?

- Do you want code that is integrated with the portal to do the authentication, or do you want to redirect the authentication responsibilities to some other program, system, or server?

- Do you want to use a supplied authentication provider? (See Table 3-10 on page 62.)

- If none of the supplied authentication provides meet your needs, do you want to develop your own authentication provider?

Worksheets

The following worksheet describes each authentication provider supplied with SIP. Go through the worksheet and indicate which authentication provider you plan to use. You can only use one authentication provider for a SIP server.

Table 3-10   Supplied Authentication Providers

Supplied Authentication Provider	Description	Plan to Use Yes/No?
Null Authentication Provider	No authentication is performed and no user name is required. Anyone who requests the portal becomes the user "anyuser."	_____
No-Password Authentication Provider	This is the default authentication provider. Requires only a user name. The user name must match the name of a user in the User Role Model (unless the defaulting mechanisms, such as a default user, are used, no authentication is performed. This provider is useful when you want to set up and try out portal views for different users.	_____
Password File Authentication Provider	Requires a user name and password. The password is stored encrypted, using UNIX `crypt(3)`. The supplied password is encrypted and compared to the stored password.	_____
LDAP Authentication Provider	Requires a user name and password. SIP connects to the LDAP server, discovers whether the user name and password can be authenticated, then disconnects from the LDAP server.	_____
Web-Server Authentication Provider	No user name and password are required if the user has already been authenticated by the web server and browser.	_____
NNM SSO Authentication Provider	No user name and password are required if the user has already been authenticated by the NNM Session Manager web login mechanism. This authentication provider requires that NNM is running on the same host as SIP.	_____

Next Steps

Continue making your plan. When you are finished:

- Use one of the supplied authentication providers by registering it with SIP (Road Map, Task 9).

- Create a custom authentication provider by developing it and then registering it with SIP (Road Map, Task 9).

The Integration of SIP Into Distributed, Wireless, and Non-English Environments

The last step in the planning process is to identify whether you plan to integrate SIP into one of the following environments: distributed, wireless, and Non-English language.

Distributed Environments

Using HTTP or HTTPS you can distribute and share the SIP configuration among multiple SIP servers. Configuration changes only need to be made in one place instead of on each SIP server, so it is easier to maintain multiple SIP servers. The SIP configuration server can be in the DMZ or behind another firewall.

Answer the following questions to help you integrate SIP into a distributed environment:

- Where will the SIP servers be deployed? In a DMZ?

- How many users will be accessing the portal?

- Will HTTPS be used to talk to the SIP servers?

- How many SIP servers will be needed to support the number of users?

- Will a web server load balancer be used?

- Will SIP configuration be distributed and shared?

- Will a firewall be used between the SIP server and the management servers?

Wireless Environment

You can configure SIP to run in a wireless environment.

- Do you want to display data through any of following modules on a PDA or cell phone?

    — NNM Network Device Health and Alarms modules

    — OVO Messages module

— Service Health and Service Cards modules

— Internet Services module

— Bookmarks and Message Board modules

Non-English Language Mode

SIP supports display of characters in UTF-8 code set to the web browser.

•   Do you want to run SIP in a non-English language mode?

Next Steps

Continue making your plan. When you are finished:

• Integrate SIP into a distributed environment by configuring entries in the SIPPath.properties file. See the whitepaper on "Configuring Service Information Portal (SIP) to Work Within a Distributed Secure Environment" (Road Map, Task 10).

• Configure SIP to deploy portals in a wireless environment. (Road Map, Task 10).

• Integrate SIP into a non-English language environment by configuring browser settings and other general settings (Road Map, Task 10).

4 Connecting SIP to Your Management Products

Configuring Management Stations to Communicate with SIP

Before you can view management data through the portal, you must establish a connection between SIP and your management servers.

For OpenView management products (NNM, OVO/OVSN, OVIS, OVR, OVPI, and OVSD), SIP provides a tool to assist you in setting up connections—the **SIP Configuration Editor**. Table 4-1 below shows how the `SIP Configuration Editor` can be used to establish these connections. The Role Properties configuration refers to the method used by OVIS, OVSD, and OVPI of associating roles with customer-to-resource mappings.

It is very important to follow the instructions in the SIP manual for each specific management product to fully configure communication with SIP.

Table 4-1   **Use of the SIP Configuration Editor by Supplied OpenView Integrations**

Product	Management Station Configuration	Role Properties Configuration
NNM	X	
OVO/OVSN	X	
OVIS	X	X
OVPM	X	X
OVR	X	
OVSD		X
OVPI		X

Starting the SIP Configuration Editor

- *Windows*: `Start:Programs->HP OpenView->Service Information Portal->Configuration Editor`
  (or from a command prompt, type: `SIPConfig`.)

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

Identifying a Host and Configuring Its Management Stations

1. In the SIP Configuration Editor, select:
   Edit:New->Management Station

2. Enter the Fully Qualified Hostname of the server on which the management software is installed. For example, host.corp.com.

3. Click [OK].

4. Select the tab for the management software installed on this host.

NOTE   The configuration settings in this dialog box accomplish part of the required steps to establish communication. For information on how to fully establish communication with SIP, you must follow the instructions in the SIP manuals that explain the individual management product integrations.

5. Click [Help] if you need assistance with the configuration settings.

6. Select another tab if you want to configure more than one management software package on this server.

7. If no other management stations are running on this host, click [OK] to return to the main SIP Configuration Editor screen, and then click [Save].

NOTE   Configuration changes are saved only when you click the [Save] button from the main SIP Configuration Editor screen, not when you click the [OK] button at the bottom of the dialog.

Establishing a Connection with NNM

The Service Information Portal runs independently from NNM, but you can configure the two applications to work together. Through SIP, NNM can provide up-to-date network information, such as alarms, maps, device status, and any requested SNMP data.

Two tasks are required to establish communication between SIP and NNM:

1. **On the SIP server, configure your NNM management servers.**

   Done through the SIP Configuration Editor, and explained in the online help.

2. **On the NNM management station, allow SIP to get information from the server.**

   Explained in "Establishing Communication Between SIP and NNM" in the manual *Network Node Manager Integration with Service Information Portal* (NNM_Integration.pdf).

Establishing a Connection with OVO and OVSN

The Service Information Portal runs independently from OpenView Operations (OVO) and OpenView Service Navigator (OVSN), but you can configure the applications to work together. Through SIP, OVO for Windows and OVO for UNIX can display up-to-date OVO messages, and OVSN can provide service management data to your customers through several OVSN modules.

Two tasks are required to establish communication between SIP and OVO:

1. **On the SIP server, configure your OVO servers.**

   Done through the SIP Configuration Editor, and explained in online help and in "Establishing Communication Between OVO and SIP" in the manual *OVO and Service Navigator Integration with Service Information Portal* (OVO_OVSN_Integration.pdf).

2. **On the OVO (for UNIX) server, define the port numbers through which SIP can communicate with OVO and Service Navigator. On OVO (for Windows) server, place a SIP-supplied script into a virtual directory on the web server where OVO Windows is running.**

Explained in "Establishing Communication Between OVO and SIP" in the manual *OVO and Service Navigator Integration with Service Information Portal* (OVO_OVSN_Integration.pdf).

Establishing a Connection with OVIS

OpenView Internet Services (OVIS) collects information about internet services and displays the data in SIP as a set of gauges and charts. SIP communicates with OVIS and requests information for display through the gauges, or it requests that OVIS generate images that SIP will present through the Internet Services module.

Three tasks are required to establish communication between SIP and OVIS:

1. **On the SIP server, configure your OVIS servers.**
   Done through the `SIP Configuration Editor`, and explained in the online help.

2. **On the OVIS measurement server.**
   Make sure OVIS is running in the appropriate mode for your version of the OVIS software (explained in "Establishing Communication Between OVIS and SIP" in *OVIS Integration with Service Information Portal* (OVIS_Integration.pdf)).

3. **On the SIP server, define the OVIS server and customer (and password if OVIS is running in secure mode) on a role-by-role basis.**
   Done through the `SIP Configuration Editor`, and explained in the online help and in the manual *OVIS Integration with Service Information Portal* (OVIS_Integration.pdf).

Establishing a Connection with a Reporting Station

The SIP Reporter modules display reports received from OpenView Reporter (OVR). In order to provide these reports, OpenView Reporter gathers data from multiple OpenView products. Communication among these products must be configured and working in order to display the information through SIP modules.

Several tasks are required to establish communication between SIP, OpenView Reporter, and the other products from which reporting data is gathered:

1. **On the SIP server, configure your reporting stations.**
   Done through the SIP Configuration Editor, and explained in
   online help and in "Establishing Communication Between OpenView
   Products for the Reporting Modules" in *OpenView Reporter
   Integration with Service Information Portal* (OVR_Integration.pdf).

2. **On the OV Reporter station, perform several configurations:**
   Establish communication with other management products from
   which you will display SIP reports. Configure OVR to generate the
   appropriate reports. Configure the integration between SIP and the
   OVR station. Perform the customer model synchronization steps. All
   of these tasks are explained in "Establishing Communication
   Between OpenView Products for the Reporting Modules" in the

   manual *OVReporter Integration with Service Information Portal*
   (OVR_Integration.pdf).

Establishing a Connection with OVSD

Several tasks are required to establish communication between SIP and
HP OpenView Service Desk (OVSD):

1. **On the SIP server, configure communication between SIP
   and Service Pages by defining OVSD servers, user names, and
   passwords on a role-by-role basis.**
   Done through the SIP Configuration Editor, and explained in the
   online help and in the manual *OVSD Integration with Service
   Information Portal* (OVSD_Integration.pdf).

2. **On each Service Desk Application Server, create a user
   account (Service Pages account) for each user that will
   access SIP Service Desk modules.**
   Refer to the manual *OVSD Integration with Service Information
   Portal* (OVSD_Integration.pdf).

Establishing a Connection with OVPI

Several tasks are required to establish communication between SIP and
HP OpenView Performance Insight (OVPI):

1. **On the SIP server, configure communication between SIP
   and Performance Insight Web Access Servers by defining
   OVPI servers, user names, and passwords on a role-by-role
   basis.**

Done through the `SIP Configuration Editor`, and explained in the online help and in the manual *OVPI Integration with Service Information Portal* (`OVPI_Integration.pdf`).

2. **On each Performance Insight Web Access Server, configure a login and password for each role that will access the SIP Performance modules, and configure a Performance Insight view for each.**
Refer to the manual *OVPI Integration with Service Information Portal* (OVPI_Integration.pdf).

5 Configuring Users and Roles

Understanding the User Role Configuration Tasks

The Service Information Portal uses an authorization model called the User Role model. By associating users with roles and defining for each role what you want the user to be able to see and do, you achieve portal security.

Users and roles are defined through a graphical user interface called the SIP Configuration Editor, the same one used to establish connections between SIP and your management software. You are encouraged to use this editor as your primary mechanism for configuring the User Role model. However, you will find that a few of the more advanced features are only configurable through direct editing of the user role XML files. Table 5-1 on page 77 lists all user role tasks and by which method they can be configured.

This chapter explains the User Role model and how to configure it through the SIP Configuration Editor and the XML files.

Table 5-1   **Required and Optional User Role Tasks and Recommended Configuration Methods**

Task	Required or Optional	Configuration Editor	XML Files
Create/modify a user role package	Required	X	
Create/modify roles	Required	X	
Create/modify users	Required	X	
Define/modify management data filters	Required	X	
Create/configure the special portal administrator role (Note: One is provided by default.**)	Required		X
Use wildcards to assign multiple roles to a user	Optional	X	
Change the name of a user role package file	Optional		X
Designate a default user	Optional		X
Designate a default role	Optional		X
Override a role's editing permissions and display name	Optional	X	
Assign a default management data filter to a user role package	Optional	X	

**SIP 2.0 Users Only: If you installed a newer version of SIP over a SIP 2.0 installation, you do not have the new default.xml and samples.xml user role package files that provide definitions of the special SIP Administrator Role and the default user. However, they are available to you if you want to copy them over your existing files or merge in the changes manually. The new files are in the following location:

Windows: %SIP_HOME%\newconfig\conf\share\roles\
UNIX: /opt/OV/SIP/newconfig/conf/share/roles/

Understanding the User Role Model

All valid portal users must be associated with at least one role. A **role** defines what a user can see and do through the portal at a particular point in time. Users that are associated with multiple roles can easily switch between them through a drop-down list in their portal.

SIP provides two types of roles: a basic role and a special portal administrator role.

Understanding Basic Roles

A basic role consists of four parts, three of which are required:

- **A portal view** (required)
  A portal view is what is displayed to a user. It is a configured set of modules and how they appear on tabs. It is also a configured set of portal view attributes, such as name in the button bar, portal skin, refresh rate, default tab, portal header, and portal footer, and so forth.

- **An editing permissions level** (required)
  Editing permission levels define the interactive editing operations that a user can perform through the portal interface. Each level of editing permissions includes all the operations defined by the previous level and some additional operations. The three levels are: ReadOnly, UserPreferences, and ViewAdmin. See "Defining a Role" on page 92 for details. For information on full administrator privileges, see "A flag that designates the role as a special SIP administrator role" on page 79 and "Creating/Configuring a Special Portal Administrator Role" on page 105.

- **A management data filter** (required)

  Management data refers to information about resources, such as services, nodes, interfaces, servers, etc., that you want to display through the portal.

  In SIP, you must filter management data at the role level, explicitly associating a filter definition with each role. Filtering on a role-by-role basis provides data security, giving you a way to ensure that only relevant data gets displayed to a given user.

You have three basic alternatives for filtering management data:

— Use the `AllData` filter to allow users of the associated role to see all data from your management stations.

— Use the `NoData` filter to allow users of the associated role to see no data from your management stations.

— Use a management data filter that specifies one or more organizations in the SIP customer model. The SIP customer model is a mapping of organizations to their resources (nodes, interfaces, and services). Only the management data associated with a specified organization is displayed, thus, segmentation of customer data can be accomplished with this third filtering alternative.

For more information on creating a customer model, see Chapter 8, "Segmenting Data by Customer Organization," on page 215.

NOTE　　　　　If you have not yet created a customer model and plan to do so later, you can use a default filter, such as "AllData" and change it later.

- **An extensible list of role properties** (optional)
  Role properties can be used to do the following types of things:

  — Display customer-segmented data from a management product that has its own customer model that is independent of the SIP customer model. (See "Understanding Management Data Filtering and Customer Data Segmentation" on page 84.)

  — Implement a single sign-on solution for logging in to back-end management applications. It can be used by modules, including the generic module. (See "Giving Your Module Access to SIP Data Through Variable Substitution" on page 180.)

Understanding Special SIP Administrator Roles

A special portal administrator role consists of only one part:

- **A flag that designates the role as a special SIP administrator role**

— If a role is flagged as "`AdminRole`", users of this role have access to the SIP Administration Pages from which the following tasks can be performed:

  — Set portal logging and tracing

  — Control the servlet engine

  — View Customer Model reports

  — Configure Customer Model refresh rates

  — Register Customer Model Sources

  — Export the Customer Model

  — Quickly log in as a different user

  — Configure messages displayed through the Message Board

  — Configure shared bookmarks

— Upon installation, SIP is configured with "sipadmin" and "admin" users. These users are associated with the SIP Administrator role, which has been designated as a special SIP administrator role. You can modify this role only through direct editing of the XML files. (See "Creating/Configuring a Special Portal Administrator Role" on page 105.)

Understanding the Supplied Users, Roles, and Views

SIP comes with several defined users, roles, and portal views. You can use these in two ways: to understand how the User Role model works, and as templates for developing your own users, roles, and views.

Figure 5-1 shows the sample users, roles, and views supplied with SIP. Each supplied user is associated with one or more roles; and each role is associated with one and only one portal view, management data filter, and edit permissions level.

Figure 5-1   **Sample User/Role/View Configurations Supplied With SIP**
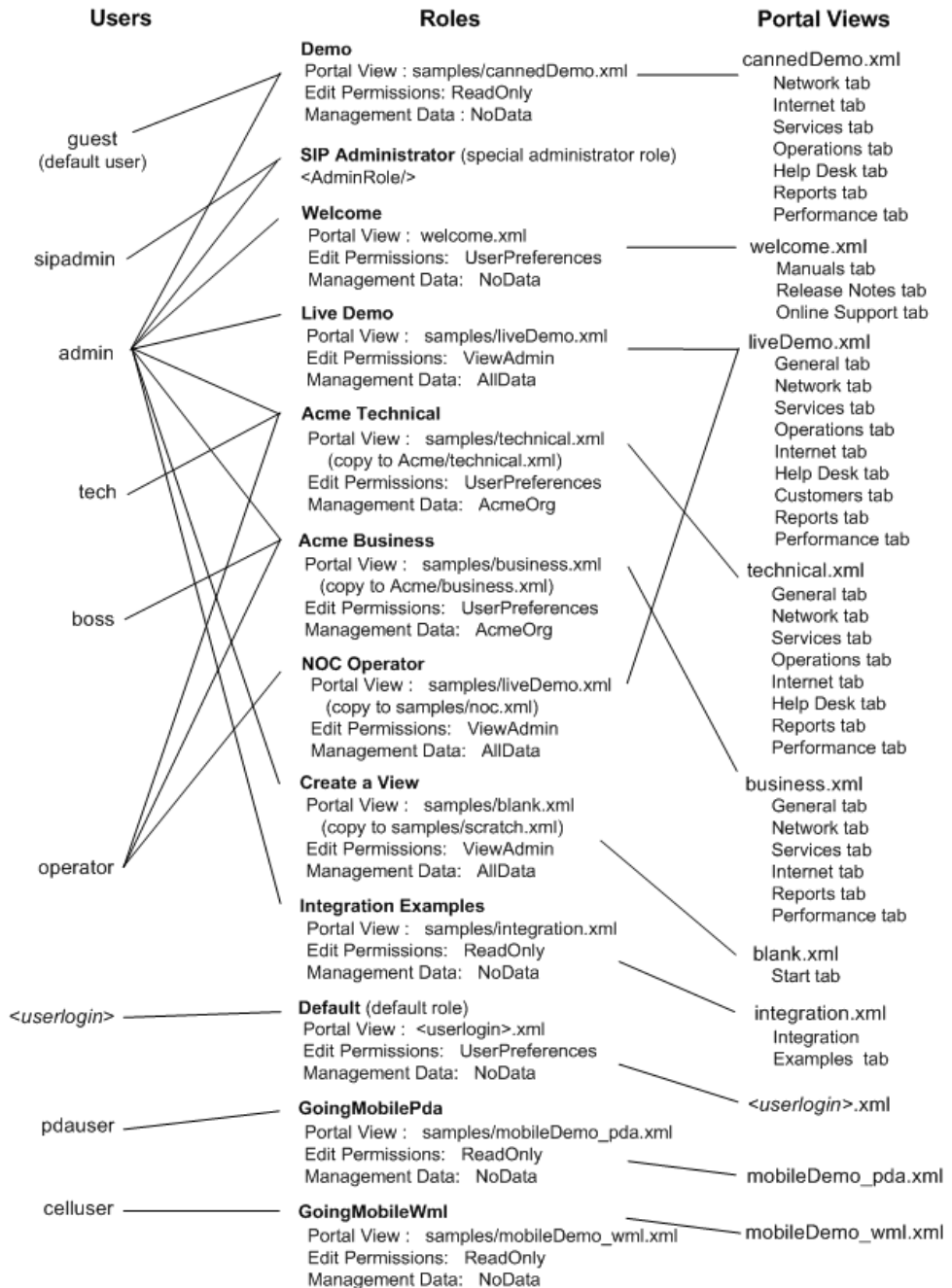
Understanding the Three Authorization Models

SIP supports three authorization models that can be enabled or disabled and used simultaneously. The three models represent a trade-off between flexibility and configuration effort.

1. **Explicit user entry**

   The user login corresponds to an explicit user entry in the User Role model. Upon login, the user gets the initial role defined in the User entry.

   This model provides the most flexibility in defining specifically what each user can see and do. This model is configured through the SIP Configuration Editor, and of the three models requires the most configuration effort. With this approach you must explicitly enter each user name into the User Role model.

2. **Default role**

   If there is no explicit user entry for the user login but there is a portal view file named *<login>*.xml and there is a role in the User Role model that is designated as the default role, upon login, the user gets the default role with the portal view file named for the login.

   You can disable this mechanism by not designating a default role. This model is configured only through the XML files. See "Designating a Default Role" on page 106.

3. **Default user**

   If neither of the two preceding mechanisms allow the user to be authorized, and if there is a user in the User Role model that is designated as the default user, upon log in the user gets the rights of the default user. The user is logged in with the initial role of the default user and can switch to any of the roles assigned to the default user. See "Designating a Default User" on page 106.

   You can disable this mechanism by not having a default user. Default user is configured only through the XML files.

Figure 5-2 is an authorization flow chart that depicts the validity checks that occur when a user logs into SIP. Note that authorization is checked in the order described above: explicit user entry, default role, and default user.

Figure 5-2   **Authorization Flow Chart**
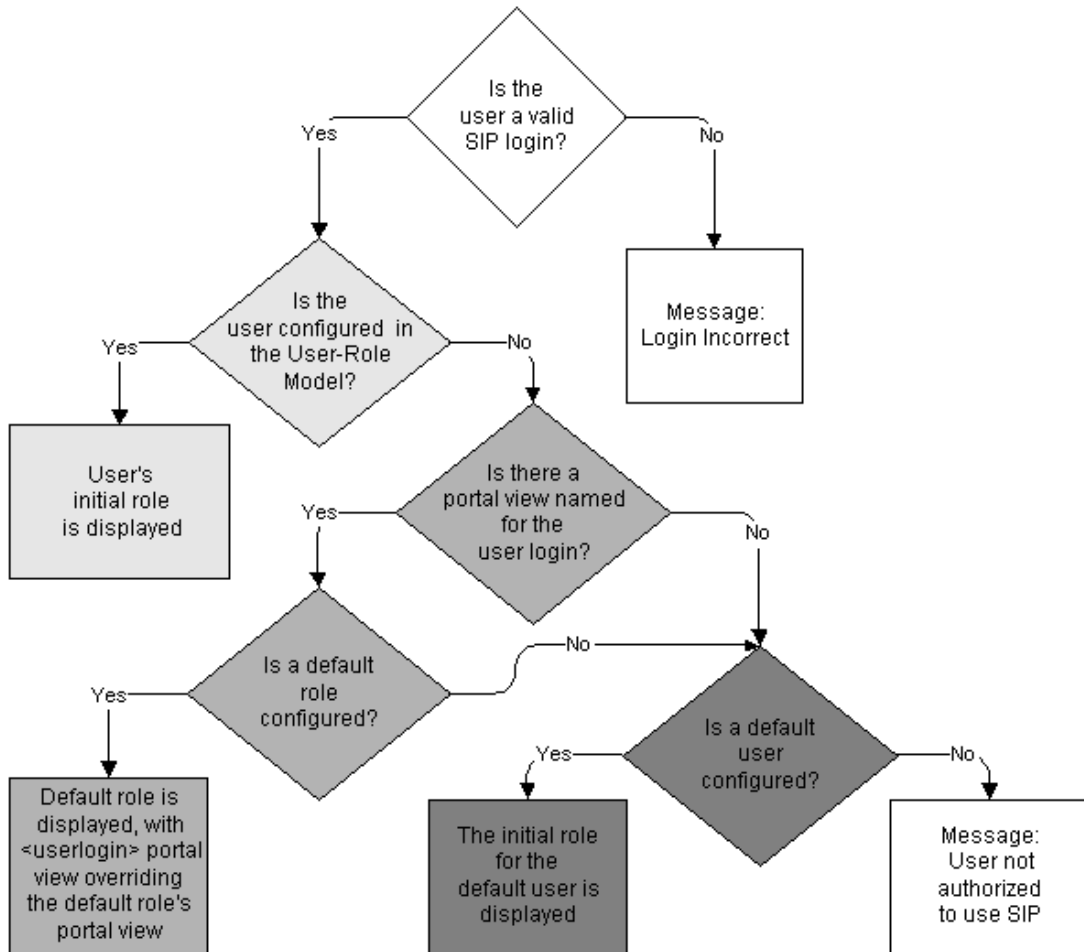
Understanding Management Data Filtering and Customer Data Segmentation

Management data refers to information about resources such as services, nodes, interfaces, servers, non-NNM-discovered objects such as containers, and so forth, that you want to display through the portal.

In SIP, you are required to filter management data at the role level. This involves explicitly associating a filter definition with each role. The filter defines what data should be displayed when a user is acting in the given role. Filtering on a role-by-role basis provides data security, giving you a way to ensure that only relevant data gets displayed to a given user.

A second level of management data filtering is allowed by some of the modules. This type of filtering is optional and allows you to refine the data on a module-by-module basis. To set up filtering at this level, see the SIP manuals that document the integration of individual management products.

Figure 5-3   **Filtering of Management Data**
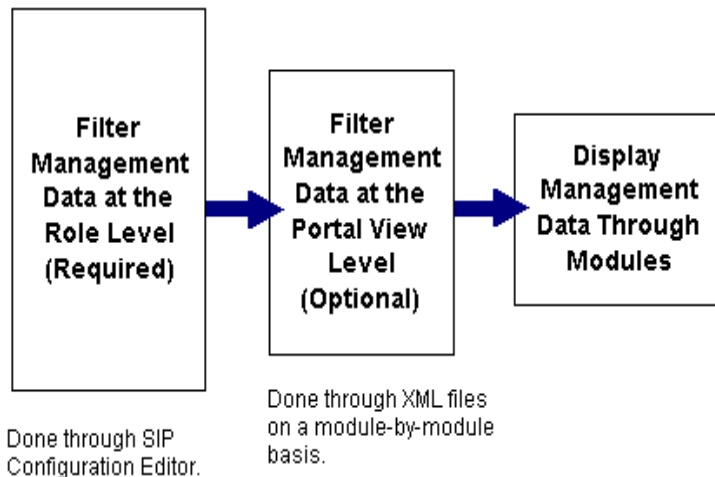
Four Choices for Filtering Management Data by Role

- Display management data that is associated with one or more customer organizations in the SIP customer model.

- Display management data from an external customer model that is maintained by a management product.

- Display all management data from the connected management stations.

- Display no management data at all.

  To display all or no data, use the SIP Configuration Editor to associate the role with a predefined management data filter: AllData or NoData.

NOTE   For more information on understanding the segmentation of customer data, see "Understanding Customer-to-Resource Mappings" on page 216.

Understanding the SIP Customer Model

Segmenting data by customer organization requires the use of a customer model: a mapping of customer organizations to their resources. Using the SIP customer model, several SIP modules can display management data that is segmented by customers.

The SIP customer model is constructed from multiple registered sources that are either XML files or programs that generate XML. The XML must be based on the SIP Simple Customer Model (SCM), which maps *organizations* to *nodes*, *interfaces*, *topomap objects* (non-NNM-discovered nodes, such as containers) and *services*, and is defined in an XML DTD.

Figure 5-4 is an example of an organization-to-resource mapping defined in a customer model source.

Figure 5-4   Organization-to-Resource Mapping in the SIP Customer Model

```
<SimpleCustomerModel>
  <Organization name="Acme" type="customer">
    <ServiceLevel>Gold Service</ServiceLevel>
    <NodeList>
       <Node name="host.acme.com"/>
       <Node name="server.acme.com"/>
    </NodeList>
    <InterfaceList>
       <Interface name="15.40.10.2" type="ov-ipv4"/>
       <Interface name="35.10.10.2" type="ov-ipv4"/>
    </InterfaceList>
    <ServiceList>
      <Service name="email" type="email">
```

```
      <DisplayString>Email</DisplayString>
      <Depth>2</Depth>
    </Service>
    <Service name="geo_orga" type="business"/>
    <Service name="cluster" type="server"/>
  </ServiceList>
  <TopomapObjectList>
     <TopomapObject type="Selection Name" name="LocationA"/>
     <TopomapObject type="Selection Name" name="LocationB"/>
     <TopomapObject type="Selection Name" name="LocationC"/>
     <TopomapObject type="Selection Name" name="LocationD"/>
  </TopomapObjectList>

  </Organization>
</SimpleCustomerModel>
```

NOTE   A word about services: The list of named services does not contain the
                  definition of the services but provides a way to refer to particular
                  services that are defined elsewhere.

For more information on understanding the SIP customer model, see
"Using the SIP Customer Model" on page 218.

Understanding External Customer Models

Other products, such as OV Service Desk, OV Performance Insight, and
OVIS, have their own customer models. SIP takes advantage of these
through special properties that can be defined for a role.

For more information on understanding external customer models, see
"Using an External Customer Model" on page 218.

Process of Using the SIP Customer Model and External
Customer Models

Table 5-2 summarizes the two alternatives for segmenting customer
data.

Table 5-2   **Summary of Alternatives for Segmenting Customer Data**

Using the SIP Customer Model	Using an External Customer Model
Create customer model sources that adhere to the SIP customer model: the mapping of *organizations* to *nodes*, *interfaces*, *non-NNM-discovered objects* (for topology maps) and *services*. Tools are provided to help you create customer model sources. Where available, they are explained in the SIP manuals that document the integration of individual management products.	Because you are using an existing customer model defined in the management product, you do not need to create a customer model. You need only assign special properties to a role in order to specify a customer organization and its resources.
After you create your customer model sources, register them with SIP. This is done through the `SIP Administration Pages` accessed through the `SIP Administrator` role.	To use an external customer model, use the `SIP Configuration Editor` to associate a role with the relevant organization and resources. This is done by defining `Role Properties` that reference a management product's existing customer model. To learn how to create valid role properties, read the SIP manuals that document the integration of these management products.
(For detailed information on creating and registering customer model sources for any of the supplied OpenView modules, see the SIP manuals that document the integration of individual management products.)	
Once the customer model sources are registered, use the `SIP Configuration Editor` to create management data filters that define the organizations to filter on. Finally, associate the filters with roles. See "Defining/Modifying Management Data Filters" on page 90.	

Configuring Users and Roles Through the Configuration Editor

The SIP Configuration Editor is a graphical user interface for configuring the User Role model. For features that are configurable only through direct editing of the user role XML files, see "Configuring Advanced Features of the User Role Model Through XML" on page 102.

In addition to giving you a way to create and modify users, roles, and management data filters, the SIP Configuration Editor provides a way to group them by customer organization. This type of grouping is called a **user role package**. Each user role package contains a set of users and roles. Combined, the packages define all the users and roles that make up the User Role model. You can define all users and roles in a single user role package, or you can partition them into multiple packages (for example, one for each customer organization).

Because all defined users, roles, and management data filters comprise a master list, each must have a unique name across all user role packages.

NOTE
SIP 2.0 Users Only: You cannot use the `SIP Configuration Editor` to modify roles that contain management data filter definitions. The definitions are still valid, but if you want to use the `SIP Configuration Editor` to modify these roles, you need to modify the XML files. For more information, see "Troubleshooting" on page 109.

Starting the SIP Configuration Editor

- *Windows*: `Start:Programs->HP OpenView->Service Information Portal->Configuration Editor`
  (or from a command prompt type: `SIPConfig`)

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

Creating/Modifying User Role Packages

1. In the `SIP Configuration Editor`:

- Select `Edit:New->User Role Package`.

- Or, select an existing user role package and choose `Edit:Properties`.

2. In the `User Role Package` dialog box, note that the `Package Name` is automatically assigned. This value becomes the name of the user role package file. SIP and the `Configuration Editor` support only ASCII file names. If you prefer a different name, see "Changing the Name of the User Role Package File" on page 107.

3. In the `Package Title` field, type a name that will help you distinguish this package from others.

4. If you have already defined management data filters and want to assign one as the default for this package, click `Assign Default Management Data` and select from the compiled list of all filters created under the various user role packages.

NOTE   Management data filters must be defined through `Edit:New->Management Data` BEFORE they can be assigned as a default management data filter. After you define the management data filter, you can return to the user role package definition and select it.

---

5. Return to the main `SIP Configuration Editor` window by clicking `[OK]`.

6. Save your configuration changes by clicking `[Save]`.

NOTE   When you save your configuration changes by clicking `[Save]` on the main `SIP Configuration Editor` window, you will receive a message titled "Results of the Save Operation." This message will indicate whether warnings and errors occurred and whether the user role database was updated.

Assigning a Default Management Data Filter to a User Role Package

For convenience while setting up roles, you can assign a default management data filter for a specific user role package and it can be used by some or all of the roles, instead of selecting a specific management data filter for each role in the package.

1. In the `SIP Configuration Editor`:

   - Select: `Edit:New->User Role Package`

   - Or highlight an existing user role package and select: `Edit:Properties`.

2. In the `User Role Package` dialog box, click `Assign Default Management Data` and select from the compiled list of all filters created under the various user role packages.

NOTE Management data filters must be defined through `Edit:New->Management Data` BEFORE they can be assigned as a default management data filter. After you define the management data filter, you can return to the user role package definition and select it.

---

3. Return to the main `SIP Configuration Editor` window by clicking `[OK]`.

4. Save your configuration changes by clicking `[Save]`.

Defining/Modifying Management Data Filters

1. In the `SIP Configuration Editor`:

   - Select the user role package (or the `Management Data` folder) to which you want to add a new management data filter, and choose: `Edit:New->Management Data`.

   - Or, select an existing management data filter and then choose: `Edit:Properties`.

2. In the `Name` field, type a name that is unique among all defined management data filters.

NOTE   The management data filter name is an internal ID used for
                            referencing the filter. If you try to use an existing filter name, when
                            you click [Save] from the main SIP Configuration Editor
                            window, you will get an error.

                        3. In the Data Presentation box, define the filter.

                            • Select Show All Data if:

                                — You want to see data through the portal but you have not yet
                                   set up a customer model (as described in Chapter 8,
                                   "Segmenting Data by Customer Organization," on page 215).

                                — You do not want to segment data by customer organization.

                            • Select Show No Data if:

                                — You do not want to display real data through the portal, such
                                   as when you are displaying demonstration modules and data.

                            • Select Show Data for the Following Organizations if you
                               have a working customer model that contains the organizations
                               that you want to filter on.

                                a. Click the [Add] button. A list of all organizations in the
                                   customer model is displayed. This list is derived from the
                                   customer model sources registered with the SIP customer
                                   model through the SIP Administration Pages (accessed
                                   through an Administrator role).

                                b. Select the organizations to filter on, and click [OK].

NOTE   If the organization name is not in the Available
                            Organizations list, you can: (1) Cancel the Select
                            Organizations dialog box and click Show All Data or Show
                            No Data. This way you can later add the organization to the
                            customer model (a good solution if you are early in the
                            development of customized portals). (2) Quit defining a
                            management data filter until you have a working customer
                            model that contains a mapping for the organization.

NOTE   If you add new organizations to the SIP customer model
                              while the SIP Configuration Editor is running, you need to
                              restart the SIP Configuration Editor before the new
                              organizations are recognized by it.

                          4. Return to the main SIP Configuration Editor window by clicking
                             [OK].

                          5. Save your configuration changes by clicking [Save].

NOTE   For information on how to overriding a role's editing permissions and
                              display name, see the next section "Overriding a Role's Editing
                              Permissions and Display Name" on page 100.

Creating/Modifying Roles

- In the SIP Configuration Editor:

  — Select the user role package folder (or the Roles folder) to which
    you want to add a new role, and choose Edit:New->Role.

  — Or, select a role and choose Edit:Properties.

NOTE   The New Role dialog box has three tabs: General, Management Data,
                              and Properties. Note that configuration changes are saved only
                              when you click the [Save] button from the main SIP
                              Configuration Editor window, not when you click the [OK] button
                              at the bottom of the dialog box.

Defining a Role

1. Select the General tab.

2. In the Role Name field, type a name that is unique across all user
   role packages.

NOTE	The Role Name is an internal ID used for referencing the role. If you try to use an existing role name, when you click [Save] from the main SIP Configuration Editor window, you will get an error.

3. In the Display Name field, type the name that will appear in the portal button bar and for the user who is acting in this role.

4. For Edit Permissions, choose from the drop-down list box:

A role must have an edit permissions level associated with it. Whereas a portal view defines what someone in a role is allowed to *see* through the portal, edit permissions defines what they are allowed to *do* through the portal—what level of editing they are allowed to perform.

- ReadOnly - A user with ReadOnly edit permissions can make no persistent editing changes. This user can make changes that have session scope, such as last tab and rollup/rolldown per module instance. This user will not see the [Options] button on the main portal page.

- UserPreferences - A user with UserPreferences edit permissions can edit the following portal attributes:

  — Display Name (in the portal button bar)

  — Skin (color Scheme and fonts)

  These changes have effect only for the user. Other users of the role will see either their own user preferences or the defaults defined in the portal view file. User preferences are saved in the file conf/share/users/*login*.xml.

  Note that because changes to display name and skin through the GUI are modified as user preferences, the default display name and skin in the portal view file can only be changed by directly editing the portal view XML file.

- ViewAdmin - A user with ViewAdmin edit permissions can perform all the operations defined for the UserPreferences level, as well as edit the following portal attributes:

  — Refresh rate

  — Show Display Name

— Show Date/Time

— Change Default tab

— Add, delete, rename, reorder tabs

— Add, delete, and reorder modules on a tab

— Customize individual modules

5. In the `Portal View` field, browse to an existing portal view, choose one, and click `Select View XML File`. You may also type the name of an existing view file, but the views file must be relative to the `conf/share/views` directory and located within the hierarchy of that directory.

NOTE   A **portal view** is a set of tabs and the configured modules that appear to a user. It also includes a set of general attributes associated with the way the portal looks to the user, such as the header, banner, "skin" (color scheme and fonts), and so forth.

Several sample portal views are supplied and can be used as templates. These are stored in the following directory:

Windows: `%SIP_HOME%\conf\share\views\samples`
UNIX: `/opt/OV/SIP/conf/share/views/samples`

To create your own custom portal views, see Chapter 6, "Customizing Portal Views," on page 111.

---

6. If you want to protect the existing portal view file (specified in the `Portal View` field) from any changes made by a user, in the `Modified View File` field enter a file name (and directory) to which a copy of the portal view file will be saved if modification are made.

For example:

Windows: `acme\technical.xml`
UNIX: `acme/technical.xml`

NOTE   `Modified View File` is an optional field for enabling multiple roles to initially share an existing portal view file. If a user makes changes to the portal view through the SIP interface, a new copy of the portal view (including the directory, if specified) will be created for the role

---

the user is currently in.

It is important to realize that you can specify only one `Modified View File` per role, and if you assign the same role to multiple users, they may overwrite each other's changes.

7. If you are finished configuring the role, return to the main `SIP Configuration Editor` window by clicking [OK], and then save your configuration changes by clicking [Save].

8. If you haven't yet assigned a management data filter, go to the `Management Data` tab explained in the next section.

Assigning a Management Data Filter to a Role

Before you can assign a management data filter to a role, you must define it through `Edit:New->Management Data`, as described in "Defining/Modifying Management Data Filters" on page 90. If you haven't yet done so, you can use one of two default management data filters supplied with SIP: `AllData` and `NoData`.

A management data filter is required for each role. Even if you are accessing data from products such as OV Internet Services (OVIS) and OV Service Desk (OVSD), which map customers to organizations through `Role Properties` and do not use the management data filter ("Assigning Special Properties to a Role" on page 97), you still must assign a management data filter.

1. Select the `Management Data` tab.

2. When assigning a management data filter to a role, you have two choices:

   • `Use Default Management Data for the Package`

      — Before you can assign default management data to a role, you must (1) define a management data filter (through `Edit:New->Management Data`), and (2) assign it as the default management data filter for the user role package (through `Edit:Properties->User Role Package`).

   • `Use Specific Management Data`

      — If you have already defined a management data filter, select it from the drop-down list in the `Data` field.

— If you have not yet defined management data filters (through `Edit:New->Management Data`), or if you do not intend to segment your management data based on customers, complete your role configuration by selecting `AllData` or `NoData` (explained in Table 5-3) from the drop-down list in the `Data` field. Later, if you create the customer model and management data filters, you can return to this tab and select a specific management data filter.

Table 5-3   When to Select AllData and NoData

Use AllData If:	**Use NoData If:**
• You want to see data through the portal but you are not yet ready to set up a customer model and management data filter for the role. • You do not want to segment data by customer organization for the role • You have not yet created a SIP customer model but plan to later. • You have created a SIP customer model but the organization name is not listed in the `Data` field (meaning that the management data filter for this customer organization has not been created).	• You do not want to display real data through the portal, such as when you are displaying canned demonstration modules and data. (In the `Default Users and Roles Package`, you will find that the `Demo` role displays `NoData`.) • For the role, you plan to display data only (1) from products that have their own customer model (OVIS and OVSD), or (2) through modules that do not display filtered customer data, such as Message Board and Bookmarks modules, and many of the Generic-based module integrations.

3. When you are finished configuring the role, return to the main `SIP Configuration Editor` window by clicking [OK], and then save your configuration changes by clicking [Save].

4. If you want to define role properties, go to the `Properties` tab explained in the next section.

Assigning Special Properties to a Role

If a management product has its own customer model, you may still be able to segment customer data through the use of Role Properties. Products such as OpenView Service Desk, OpenView Internet Services, and OpenView Performance Insight have their own customer models that SIP can take advantage of.

Another way to use Role Properties is when you are developing Generic-based modules. You can extend the module in a way that allows the module to retrieve user-specific information from SIP. In this way, your application can get data about a user's current role, edit permissions, organizations defined by the management data filter, and properties. This is done through a predefined variable: $OVROLE. For more information on using Role Properties this way, see "Giving Your Module Access to SIP Data Through Variable Substitution" on page 180.

To Define Role Properties

1. Click on the Properties tab.

2. Enter both a Name and Value that are understood by the management product from which segmented data is to be gathered. To ensure that you are entering valid properties, you must follow the instructions in:

   - *OVIS Integration with SIP* (OVIS_Integration.pdf)

   - *OV Service Desk Integration with SIP* (OVSD_Integration.pdf)

   - *OV Performance Insight Integration with SIP* (OVPI_Integration.pdf)

   - *OV Performance Manager with SIP* (OVPM_Integration.pdf)

3. Click [Add].

4. When you are finished configuring the role, return to the main SIP Configuration Editor window by clicking [OK].

5. Save your configuration changes by clicking [Save].

To Modify Role Properties

1. Click on the Properties tab.

2. In the Defined Role Properties table, double-click in the cell you want to modify.

3. Type the changes.

4. When you are finished configuring the role, return to the `main SIP Configuration Editor` window by clicking `[OK]`.

5. Save your configuration changes by clicking `[Save]`.

To Remove Role Properties

1. Click on the `Properties` tab.

2. If you want to remove properties from this role, select a row in the `Defined Role Properties` box and click `[Remove]`. Multiple properties can be selected simultaneously using `CTRL+Click` and `SHIFT+Click`.

3. Multiple role properties can be selected in the table and removed simultaneously using `CTRL+Click` and `SHIFT+Click`.

4. When you are finished configuring the role, return to the main `SIP Configuration Editor` window by clicking `[OK]`.

5. Save your configuration changes by clicking `[Save]`.

Creating/Modifying Users

- In the `SIP Configuration Editor`:

  — Select the user role package folder (or the `Users` folder) to which you want to add a new user, and choose `Edit:New->User`.

  — Or, select a user and choose `Edit:Properties`.

NOTE   The `New User` dialog box has two tabs: `General` and `Roles`. Note that configuration changes are saved only when you click the `[Save]` button from the main `SIP Configuration Editor` window, not when you click the `[OK]` button at the bottom of the dialog box.

Defining a User

1. Click on the `General` tab.

2. In the `User Login Name` field, type a name that is unique across all user role packages. User names must be valid SIP login names registered with the authentication provider.

NOTE   `User Login Name` is an internal ID used for referencing a user. If you try to use an existing user login name, when you click [Save] from the main `SIP Configuration Editor` window, you will get an error.

---

3. In the `Display Name` field, type the name that will appear to the user in the portal button bar.

Assigning Roles to a User

After you have defined roles, you can assign to your users the roles you want them to have. Users can have multiple roles between which they can switch while using the portal. Users must be associated with at least one role: the initial role.

1. Click on the `Roles` tab.

2. From the drop-down list in the `Initial Role` field a list of all defined roles is displayed. Select a role that the user will be in when he or she first logs in to the portal.

3. To assign more than one role to this user, click [Add]. A list of all defined roles (in all user role packages) is displayed.

   In the `Select Role` dialog box, you can sort the list of roles by any column by double-clicking on the column heading.

4. Select a role. Multiple roles can be selected simultaneously using `CTRL+Click` and `SHIFT+Click`. Once you have the desired roles selected, click [OK].

5. When finished, click [OK] to return to the main `SIP Configuration Editor` window.

6. Save your configuration changes by clicking [Save].

Assigning a Set of Roles to a User Using Wildcards

If you have several roles that begin with the same name, for example, Acme, you can use wildcards to assign all Acme roles to a user.

1. On the Roles tab, first add a row to the Other Roles table by clicking [Add], choosing any of the roles, and clicking [OK].

2. In the Name field, delete the existing name, and type the name and wildcard that defines a set of roles to reference. For example, to select all roles that begin with Acme (such as AcmeTechnical and AcmeBusiness, type Acme.*.

3. To prepend a name to each role, for example, Acme Organization, in the Override Display Name field type Acme Organization followed by a space. The name of the role as it appears in the role drop-down list on the portal button bar will be prepended with "Acme Organization". For example, "Acme Organization Technical."

4. When you are finished configuring the user roles, return to the main SIP Configuration Editor window by clicking [OK].

5. Save your configuration changes by clicking [Save].

Overriding a Role's Editing Permissions and Display Name

You can allow a user to switch to another user role and be granted a different permission level than the default for the role. You can do this for any user, but it is recommended that you set a higher level only for administrative users.

1. In the SIP Configuration Editor, select a user and select: Edit:Properties.

2. Click on the Roles tab.

3. If the role is not yet assigned to this user, in the Other Roles segment of the dialog box, click [Add].

4. A list of all roles with current display names and permission levels appears. Select one or more and click [OK].

5. On the Roles tab, find the role for which permissions will be overridden, and click in the Override Permissions column. From the drop-down list, select a different level of permissions than the default.

6. You can also override the Display Name of the role. The Display Name appears in the portal button bar to the current user acting in this role. On the Roles tab, for a given role double-click the Override Display Name cell. Type the new name.

7. When you are finished configuring the override, return to the `main SIP Configuration Editor` window by clicking `[OK]`.

8. Save your configuration changes by clicking `[Save]`.

Verifying that a Portal Works as Expected

After you have configured the users and roles, you can start SIP and switch among roles to verify that each is configured correctly.

1. Log into SIP as "admin" or "sipadmin" or some other administrative user.

2. Switch to the `SIP Administrator` role. (If an administrator role does not appear on the list of roles, you may need to log in as a different administrative user.)

3. Click on the `Switch Users` tab.

4. Click on the user name in the list box and select `[Become User]`.

5. Examine the portal to verify that it looks and works as expected.

6. To switch back to the original user, click the `[Logout]` button.

Configuring Advanced Features of the User Role Model Through XML

Several user role configuration tasks can only be done through the XML files:

- Configuring a special portal administrator role

- Designating a default user

- Designating a default role

- Changing the name of the user role package file

Procedures for these tasks are explained on the following pages. For more information on the XML elements and attributes, see Appendix E, "Configuring Basic Features of the User Role Model Using XML," on page 321.

NOTE	**SIP 2.0 Users Only:** If you installed SIP 3.0 over your SIP 2.0 installation you do not have the new `default.xml` and `samples.xml` user role package files that provide definitions of the special SIP Administrator Role and the default user. However, they are available to you if you want to copy them over your existing files or merge in the changes manually. The new files are in the following location:

Windows: `%SIP_HOME%\newconfig\conf\share\roles\`
UNIX: `/opt/OV/SIP/newconfig/conf/share/roles/`

Location of the XML Files

The user role files are located in the following directory:

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

What Gets Installed With SIP

Figure 5-5 on page 103 shows the directory structure when SIP is installed. Here is an explanation of each file:

- `default.xml` - A package configuration file that defines a special administrator role, as well as a default user and default role used when authorizing valid SIP users who are not explicitly defined in the User Role model. It also contains the definition of `AllData` and `NoData` management data filters.

- `index.xml` - Used as an index to all package files that contain user and role information and comprise the User Role model.

- `package.xml` - A package file that you can copy and rename as a starting point when creating your own package files.

- `README` - Important notes on updating the User Role model when directly modifying the XML.

- `samples.xml` - A package configuration file that defines several sample users and roles for out-of-the-box use of SIP.

- `UserRole.dtd` - Defines the XML DTD for the user role packages. For detailed information about the elements and attributes of UserRole.dtd, see "User Role Model Elements and Attributes" on page 327.

Figure 5-5   **roles directory**

```
roles
    default.xml
    index.xml
    package.xml
    README
    samples.xml
    UserRole.dtd
.db
    default    roles    users
```
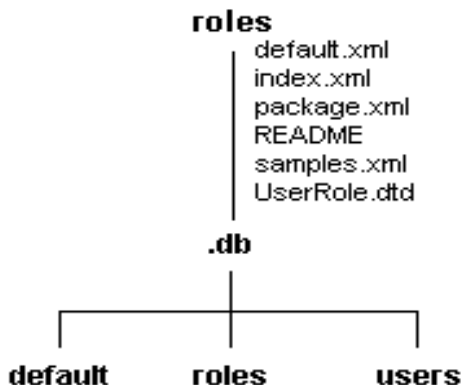
Understanding the User Role XML Files

- Two types of configuration files make up the User Role model: an index file and user role package files.

- You can store the user role package files in any directory structure under the `roles` directory. Do not, however, put any files in the generated `.db` database directory.

- Never edit any file in the `.db` directory or its subdirectories. These files constitute the user role database and are automatically generated by the `create_role_db` command or when the `[Save]` operation in the `SIP Configuration Editor` completes without errors.

- The index file registers all of the package files that make up the User Role model. The index file defines the `UserRoleModel` element, shown in Figure E-1 on page 327 and described in Table E-3 on page 329. You can have only one index file and it must be named `/conf/share/roles/index.xml`.

- Each user role package file contains a set of users and roles. Combined, the package files define all the users and roles that make up the User Role model. You can define all users and roles in a single file or you can partition them into multiple files (for example, one for each customer organization). User role package files must be referenced in the index file.

- Before you edit the XML files, make sure the `SIP Configuration Editor` is closed.

- After you make changes to the user role configuration files, you must (1) run the `create_role_db` command or (2) start the `SIP Configuration Editor` and perform a successful `[Save]` operation to update the database files. For more information, see below.

Updating the User Role Model After Editing the XML Files

If you make changes to the User Role model through direct editing of the XML files, you must update the User Role database in one of two ways:

- Start the `SIP Configuration Editor` and perform a successful `[Save]` operation, or

- Run the command `create_role_db`

At run-time, the SIP portal uses the user role data from the database not from the User Role model XML files. If the User Role model contains no errors, the database is updated and the changes are immediately available to the portal.

To update the database from the command line:

1. First, make sure the SIP Configuration Editor is closed.

2. After you have updated the user role XML files, run
   create_role_db.

NOTE   For the command to work from outside the bin directory, you must
set JAVA_HOME to the JDK 1.3 environment, and add the following to
your PATH variable:

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

3. Correct any problems that are detected, and run the command until
   the User Role model is satisfactory.

Creating/Configuring a Special Portal Administrator Role

You can give to a role the highest level of administrative privileges by
defining the AdminRole element in the role. You should have at least one
role designated as the SIP administrator role. For more information, see
"Understanding Special SIP Administrator Roles" on page 79.

Upon installation, the special SIP administrator role is assigned to the
admin user in the samples.xml package file and the sipadmin user in
the default.xml user role package. You can designate more than one
administrator role, but be cautious. The presence of this element grants
access to the SIP Administration Pages as well as the ability to log in
as any user.

Creating the special Administrator role can be done through the direct
editing of XML files. You cannot create it through the SIP
Configuration Editor. However, once it is created, you can use the SIP
Configuration Editor to assign it to users.

The following role configuration is provided for you in the default.xml
user role package file:

```
<!-- Special SIP Administrator Role -->

<Role name="SIPAdministrator"
      title="SIP Administrator">
      <AdminRole/>
</Role>
```

Designating a Default User

Upon installation, SIP provides a definition for a default user. The default user can be modified, but not set, through the SIP Configuration Editor. Only one default user can be defined in the User Role model. For important information on the default user feature, see "Understanding the Three Authorization Models" on page 82.

The following definition of a default user is provided in the default.xml user role package file.

```
<User name="guest" displayName="Guest"
      initialRole="Demo" defaultUser="yes">
   <RoleRef href="Demo"/>
</User>
```

To Set the Default User

1. To remove the default user designation from a user (for example, from the user "guest" shown above), change the defaultUser attribute from "yes" to "no", or remove this attribute all together.

2. To designate a different user as the default user, in the User element, define the defaultUser="yes".

3. Save the user role package file to which changes were made, and follow "Updating the User Role Model After Editing the XML Files" on page 104.

Designating a Default Role

Upon installation, SIP provides a definition for a default role. The default role can be modified, but not set, through the SIP Configuration Editor. Only one default user can be defined in the User Role model. For more information on the default role feature, see "Understanding the Three Authorization Models" on page 82.

The following definition of a default user is provided in the default.xml user role package file.

```
<Role name="Default" title="Default" defaultRole="yes">
<!-- The portal view file "<login>.xml" may override this -->

   <PortalViewRef href="samples/cannedDemo.xml"/>
   <EditPermission level="UserPreferences"/>
   <MgmtDataRef href="NoData"/>
</Role>
```

To Set the Default Role

1. To remove the default role designation from a role (for example, from the role "Default" shown above), change the defaultRole attribute from "yes" to "no", or remove this attribute all together.

2. To designate a different role as the default role, in the Role element, define the defaultRole="yes".

3. Save the user role package file to which changes were made, and follow "Updating the User Role Model After Editing the XML Files" on page 104.

Changing the Name of the User Role Package File

When you create a user role package in the SIP Configuration Editor, the Package Name field is automatically assigned and defines the name of the user role package file on the file system. SIP and the Configuration Editor support only ASCII file names. If you prefer to change the package file name, follow these steps:

1. In the roles directory, find and rename the user role package file:

   Windows: %SIP_HOME%\conf\share\roles\
   UNIX: /opt/OV/SIP/conf/share/roles/

NOTE   Package file names must contain only ASCII characters.

---

2. Change the package file name in the index file:

   Windows: %SIP_HOME%\conf\share\roles\index.xml
   UNIX: /opt/OV/SIP/conf/share/roles/index.xml

   For example:

   `<UserRolePackageRef href="`*`<modified_package_name>`*`.xml"/>`

3. Save and close the `index.xml` file.

4. Update the User Role model by following the instructions in "Updating the User Role Model After Editing the XML Files" on page 104.

Troubleshooting

Role Contains Management Data Filter Not Editable Through SIP Configuration Editor

This problem is experienced by SIP 2.0 Users Only. A role that contains a management data definition (instead of a reference to a management data definition) cannot be edited through the SIP Configuration Editor. The definitions are still valid, but if you want to use the editor to modify these roles, you will need to use an XML editor to modify the role, removing the management data filter definition from the role and instead creating it outside the role and referencing it from the role.

Figure 5-6   **Sample Management Data Definitions Defined Inside a Role**

```
<Role name="AcmeBusiness" title="Business">
   <PortalViewRef href="samples/business.xml"
                   copy="Acme/business.xml"/>
   <EditPermission level="UserPreferences"/>
   <MgmtData name="AcmeData" orgName="Acme"/>
</Role>

<Role name="BestCorpManager" title="Business">
   <PortalViewRef href="samples/business.xml"
                   copy="BestCorp/business.xml"/>
   <EditPermission level="UserPreferences"/>
   <MgmtData name="BestCorpBusiness">
           <OrganizationFilter op="OR">
             <OrganizationRef href="BestCorpFinance"/>
             <OrganizationRef href="BestCorpHR"/>
           </OrganizationFilter>
   </MgmtData>
</Role>
```

Figure 5-7   **Sample Management Data Definitions Defined Outside the Role and Referenced from the Role**

```
<Role name="AcmeBusiness" title="Business">
   <PortalViewRef href="samples/business.xml"
                   copy="Acme/business.xml"/>
```

```
      <EditPermission level="UserPreferences"/>
      <MgmtDataRef href="AcmeData"/>
   </Role>

   <Role name="BestCorpTechnical" title="Technical">
      <PortalViewRef href="samples/technical.xml"
                     copy="BestCorp/technical.xml"/>
      <EditPermission level="UserPreferences"/>
      <MgmtDataRef href="BestCorpBusiness">
   </Role>

   <MgmtData name="AcmeData" orgName="Acme"/>

   <MgmtData name="BestCorpBusiness">
      <OrganizationFilter>
         <OrganizationRef href="BestCorpFinance"/>
         <OrganizationRef href="BestCorpHR"/>
      </OrganizationFilter>
   </MgmtData>
```

6 Customizing Portal Views

Understanding Portal Views

A portal view is the configured set of tabs and modules that appear to a user. It is also a configured set of portal view attributes, such as the name in the button bar, the portal skin, refresh rate, default tab, portal header, and portal footer, and so forth.

A portal view is that part of a role that defines what a user can see through the portal at a particular point in time.

SIP comes with several defined portal views. You can use them as templates for developing your own views. For an illustration of the supplied portal views, users, and roles, see Figure 5-1 on page 77.

The task of creating and customizing portal views means first deciding what modules, tabs, and portal view attributes should be associated with a given role, whether you can leverage the supplied portal views, and whether you want to customize the portal look and feel by using your own graphic images in the portal banner and footer. Before beginning to customize portal views, make sure you have made a plan. See Chapter 3 "Questions to Ask Yourself" on page 49 and the worksheets in Table 3-6 on page 51 and Table 3-7 on page 52.

Customization Options

The following two tables list the basic and advanced customizations that you can make to portal views.

Basic portal view customizations are made through the portal interface while you are in a role that references the portal view that you want to customize. Advanced customizations are made through the direct editing of XML, HTML, and JSP files.

Table 6-1   Basic Portal View Customizations

Customization	Configuration Method (through SIP interface)	Minimum Edit Permissions Required of the Role
Specify the User Name to Display in the Button Bar of a Portal View	[Options] button	UserPreferences
Choose the Skin (Portal Look and Feel) for a Portal View	[Options] button	UserPreferences
Choose the Refresh Rate for a Portal View	[Options] button	ViewAdmin
Choose to Show a User Name in the Button Bar of the Portal View	[Options] button	ViewAdmin
Choose to Show the Date and Time in the Button Bar of the Portal View	[Options] button	ViewAdmin
Create, Modify, Delete, and Reorder Tabs in a Portal View	[Options] button	ViewAdmin
Select the Default Tab for a Portal View	[Options] button	ViewAdmin

Table 6-1   Basic Portal View Customizations (Continued)

Customization	Configuration Method (through SIP interface)	Minimum Edit Permissions Required of the Role
Add Modules to a Portal View	`[Options]` or `[Add]` button at the bottom of each tab column	`ViewAdmin`
Remove Modules from a Portal View	`[X]` button on the module title bar. `[Options]` button, or `[Edit]` button at the bottom of each tab column	`UserPreferences`
Reorder Modules in a Portal View	`[Options]` button, or `[Edit]` button at the bottom of each tab column.	`ViewAdmin`
Add, Remove, and Reorder Submodules in a Portal View	`[Edit]` button on a module title bar.	`ViewAdmin`

Table 6-2   Advanced Portal View Configurations

Customization	Configuration Method (direct editing of XML files)
Change the Default Help Topic for a Module	Direct Editing of module registration file
Change the Default Header for all portal views	Direct Editing of `OVPortalConfig.xml`
Change the Default Footer for all portal views	Direct Editing of `OVPortalConfig.xml`
Specify a Custom Help Topic for a Specific Portal View	Direct Editing of the portal view XML file
Specify a Custom Header for a Specific Portal View	Direct Editing of the portal view XML file

Table 6-2   **Advanced Portal View Configurations (Continued)**

Customization	Configuration Method (direct editing of XML files)
Specify a Custom Footer for a Specific Portal View	Direct Editing of the portal view XML file
Add a Custom Skin to the Skin Selection List for all portal views	Direct Editing of `OVPortalConfig.xml`
Create a Custom Help Topic for a Module	HTML
Create a Custom Header	JSP
Create a Custom Footer	JSP
Create a Custom Skin	Cascading Style Sheets (CSS)
Configure module display filters	Direct Editing of the portal view XML file. For instructions on configuring module display filters, see the SIP manuals that explain the integrations of individual management products: "Getting Additional Documentation" on page 17.
Configure advanced module-specific settings	Direct Editing of the portal view XML file For instructions on configuring advanced module-specific settings, see the SIP manuals that explain the integrations of individual management products: "Getting Additional Documentation" on page 17.

Creating Portal Views

An efficient way to develop portal views is to leverage existing ones. Following are two ways to create a new portal view file based on one that is supplied or on one that you have created:

- From the file system:

    1. Copy and rename an existing portal view XML file (one that is supplied with SIP or already created by you). Portal view files are located in the following directory:

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

        If you change the directory levels when copying a portal view file, you must update the DTD path in the file.

        In each portal view file is a reference to the `PortalView.dtd`. This DTD is specified relative to the `conf/share/views` directory. If your portal view file is located in the `conf/share/views` directory, the DTD reference looks like this:

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

        If you copy portal view files between the `conf/share/views` directory and any of its subdirectories, make sure you change the reference to the `PortalView.dtd`. For example, if you place a portal view file in the `conf/share/views/samples` directory, the reference to the DTD would be:

        `<!DOCTYPE PortalView SYSTEM "../PortalView.dtd">`

    2. Through the `SIP Configuration Editor`, create or modify a role and assign the new portal view, that you created in step 1, to the role.

- From the `SIP Configuration Editor`:

    — Create or modify a role and assign an existing portal view, and a modified view file, to the role. (For an example, see.) When the portal view is later customized, the original portal view is protected while a new copy is created with the modifications.

In this way, you can enable multiple roles to initially share a portal view, and when changes are made to the portal view through the SIP interface, a new copy of the view is created for the role. This protects the initial portal view file from any changes made by a user.

It is important to note that if you make changes to the original portal view, those changes are NOT made automatically in the modified view file. In addition, you can specify only one modified view file per role. If you assign the same role to multiple users, they may overwrite each other's changes.

Figure 6-1   Sample Entry of a Modified View File
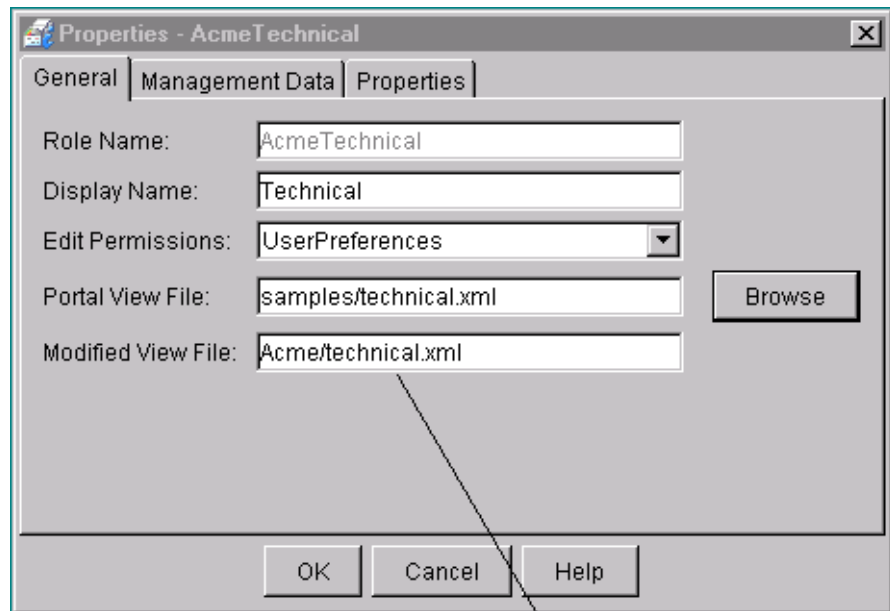
Modified View File

For detailed information on creating modified view files, see "Creating/Modifying Roles" on page 92.

Logging In to SIP to Customize a Portal View

- Log in to SIP as a user who has access to the role that references the portal view you want to customize. If the user has access to multiple roles, switch to the appropriate role.

You can log in as either an end user or an administrative user who has access to the role. To edit a portal view, the role must have the appropriate level of permissions. The majority of the customizations require the role to have `ViewAdmin` edit permissions. Table 6-1 on page 113 shows the permissions level required to make each customization. For information on overriding editing permissions, see "Overriding a Role's Editing Permissions and Display Name" on page 100.

Giving an Administrative User "ViewAdmin" Permissions to All Roles

You can give permission to an administrative user to customize any portal view by giving them access to all roles and overriding the edit permissions level.

1. In the `SIP Configuration Editor`, select a user and choose `Edit:Properties`.

2. Click on the `Roles` tab.

3. In the `Other Roles` segment of the dialog box, click `[Add]`.

4. To add a blank row to the list in the `Other Roles` list, choose any role on the list, and click `[OK]`. It doesn't matter which role you select because you will change the values.

5. On the `Roles` tab, change the role `Name` to the wildcard `".*"` as shown on the next page.
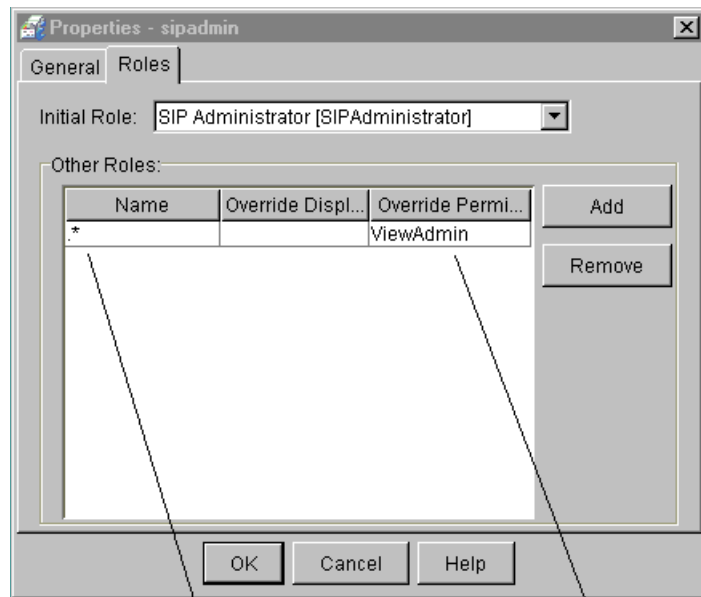
wildcard value of " .* "
matches all roles

Edit permissions override
value of "ViewAdmin"
gives sipadmin user that
level of permissions for all
roles

6. In the `Override Permissions` column, from the drop-down list
   select `ViewAdmin`.

7. Return to the `main SIP Configuration Editor` window by clicking
   `[OK]`.

8. Save your configuration changes by clicking `[Save]`.

Customizing General Portal View Attributes

The following portal view attributes can be configured in order to personalize a portal view:

- User name that appears in the button bar

- The portal skin (portal look and feel)

- Portal refresh rate

- A switch to allow for the display of the user name in the button bar

- A switch to allow for the display of the date and time in the button bar.

Specifying the User Name to Display in the Button Bar of a Portal View

The role you are in must have a minimum edit permissions level of UserPreferences in order to change the name that appears in the button bar.

1. In the main portal page, click [Options] on the portal button bar.

2. In the User Options segment, type the name that you want to display in the portal button bar.

3. Save and return to the main portal page by clicking [OK].

Choosing the Skin for a Portal View

The role you are in must have a minimum edit permissions level of UserPreferences in order to change the portal skin.

1. In the main portal page, click [Options] on the portal button bar.

2. Go to the User Options segment of the page.

3. From the Skins drop-down list box, select a skin that provides the look and feel you want the portal to have. Try the available skins to find one that you like.

4. Save and return to the main portal page by clicking [OK].

Choosing the Refresh Rate for a Portal View

Refresh rate refers to the frequency (by default, in minutes, hours, or one day) at which the data on the portal page is regenerated to reflect the most current state of the data. Be aware that frequent refreshes can decrease portal performance.

The role you are in must have a minimum edit permissions level of `ViewAdmin` in order to change the refresh rate.

1. In the main portal page, click `[Options]` on the portal button bar.

2. Go to the `Portal Options` segment of the page.

3. From the Refresh Page Every drop-down list box, select a frequency that will determine the time interval between page updates.

4. Save and return to the main portal page by clicking `[OK]`.

Choosing to Show a User Name in the Button Bar

The role you are in must have a minimum edit permissions level of `ViewAdmin` in order to choose whether or not to show a user name in the portal button bar.

Choosing to Show the Date and Time in the Button Bar

The role you are in must have a minimum edit permissions level of `ViewAdmin` in order to choose whether or not to show the date and time in the portal button bar.

Customizing Tabs and Modules in a Portal View

This section describes how to create and customize tabs and modules for a portal view.

Configuring Tabs

You can create any number of tabs and customize them in a variety of ways, all depending upon how you want to present information through a portal view.

The role you are in must have a minimum edit permissions level of ViewAdmin in order to configure the portal view tabs.

1. Log in to SIP as a user who has access to the role that references the portal view you want to customize. Switch to the role, if necessary.

2. On the main portal page, click [Options] on the portal button bar.

3. Configure the tabs, as described in the following sections:

Adding a New Tab

When you add tabs to a portal they are appended to the right of existing tabs.

1. On the Options page, go to the Tab Settings segment.

2. Click [Add].

3. On the Add New Tab page, type the name of the new tab.

4. In the Columns segment, use the buttons to add any number of narrow or wide columns. By default, one narrow and one wide column are added.

5. Change the order of the columns by selecting one and clicking [Move Left] or [Move Right].

6. To save the new tab and display it so you can add modules, click [Go to Tab].

7. Or, to save and return to the Options page, click [Return to Options] and then click [OK] to return to the main portal page.

Choosing a Default Tab

1. On the Options page, go to the `Tab Settings` segment.

2. In the `Default Tab` field, select the tab from the drop-down list box. This is the tab that is displayed when a user selects this role.

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

Modifying a Tab

1. On the Options page, go to the `Tab Settings` segment.

2. Select a tab and click `[Modify]`.

3. On the `Modify Tab` page, change the name of the tab, if desired.

4. In the `Columns` segment, use the buttons to add, remove, or modify columns. If a column is removed, all the modules within it are also removed.

5. To save the changes and display the tab, click `[Go to Tab]`.

6. Or, to save and return to the Options page, click `[Return to Options]` and then click `[OK]` to return to the main portal page.

Deleting a Tab

1. On the Options page, go to the `Tab Settings` segment.

2. Select a tab and click `[Delete]`.

3. When prompted, confirm the delete action.

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

Reordering a Tab

1. On the Options page, go to the `Tab Settings` segment.

2. Select the tab that you want to reorder, and click `[Move Right]` or `[Move Left]`.

3. Save and return to the main portal page by clicking `[OK]`.

Configuring Modules

You can add any number of modules to a tab and customize them in a variety of ways, all depending upon how you want to present information through a portal view. Some modules can only be added to wide columns.

The role you are in must have a minimum edit permissions level of ViewAdmin in order to configure the portal view tabs.

1. Log in to SIP as a user who has access to the role that references the portal view you want to customize. Switch to the role, if necessary.

2. On the main portal page, select the tab on which you want to configure modules.

3. Configure the modules, as described in the following sections:

Adding Modules to a Portal View

When you add modules to a portal view, you are adding the module as it is configured by default and registered with SIP. Default module configurations are stored in the following directory:

Windows: %SIP_HOME%\registration\defaults
UNIX: /opt/OV/SIP/registration/defaults

A fast and convenient way to add modules is through the module selection list at the bottom of each tab column. You can add one module to the tab, or add several modules at a time.

- Adding One Module

    1. Scroll to the bottom of the tab to which you want to add a module.

    2. Open the drop-down list box below the column, select the module you want to add, and click [Add].

- Adding Several Modules at a Time

    1. Scroll to the bottom of the tab to which you want to add modules.

    2. Go to the bottom of the column and click [Edit].

    3. On the Modify Column page in the Available Modules list, select a module and click [Add].

    4. To save the changes and return to the tab, click [Go to Tab].

- Creating Tabs and Adding Modules

Another way to add modules is through the [Options] button on the main portal page. This is particularly useful if you are setting up a tab from start to finish, including adding modules.

Modifying Modules in the Portal View

You can easily modify modules and customize them in a variety of ways, all depending upon how you want to present information through a portal view.

The role must have a minimum edit permissions level of ViewAdmin in order to configure the portal view tabs.

1. Log in to SIP as a user who has access to the role that references the portal view you want to customize. Switch to the role, if necessary.

2. On the main portal page, navigate to the tab on which the module appears.

3. Go to the module and click the edit button on the module title bar:

NOTE   For assistance on the individual module edit pages, click [Help]

4. To add a submodule to the portal view, select it from the list of Available *Submodules*, and click [Add].

5. To remove one from the portal view, select it from the list of Displayed Submodules, and click [Remove].

6. Some modules provide for the reordering of submodules. To reorder one, select it and click [Up] or [Down].

7. Some modules have additional configuration settings on the edit page or through direct editing of the XML files. Refer to the SIP manuals that document the individual management product integrations for detailed information.

NOTE   Some modules provide for configurations that are made at the global level. The settings affect all instances of a given module in every portal view to which it is added. Other configurations can be made on a per-module instance basis, affecting only the module as it appears

in a particular portal view. Furthermore, for some modules you can configure a filter at the module instance to further refine the management data that is displayed through the portal view. This type of filtering, sometimes referred to as display filtering, further restricts, beyond the management data filter, what the user sees through the portal view.

Refer to the SIP manuals that document the individual management product integrations for detailed information.

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

Removing Modules from a Portal View

When you remove a module, the action is final. Configurations made to the module cannot be restored. You can only add the default (unconfigured) module, as described in "Adding a New Tab" on page 122.

NOTE   You can use the rollup/rolldown feature if you only want to temporarily disable a module and not remove it. For more information, see "Restoring Modules That Have Been Removed" on page 127.

You can remove one module from the tab or remove several at a time.

- Removing One Module at a Time

  1. Navigate to the tab that has the module you want to remove.

  2. On the module title bar, click the [X] button.

- Removing Several Modules at a Time

  1. Scroll to the bottom of the tab to which you want to remove modules.

  2. Go to the bottom of the column and click [Edit].

  3. On the Modify Column page in the Available Modules list, select a module and click [Remove].

  4. Repeat step 3 until you have removed all of the modules you want to remove.

  5. To save the changes and return to the tab, click [Go to Tab].

Changing the Display Order of Modules

1. On the tab, go to the bottom of the column and click [Edit].

2. On the Modify Column page in the Available Modules list, select a module and click [Up] or [Down].

3. To save the changes and return to the tab, click [Go to Tab].

Restoring Modules That Have Been Removed

If you removed a module by clicking the close button [X] on the module title bar, the module cannot be restored. If you temporarily disabled a module instance by clicking the rollup button (Figure 6-2), you can restore the module by clicking the rolldown button (Figure 6-3).

Figure 6-2   Rollup button
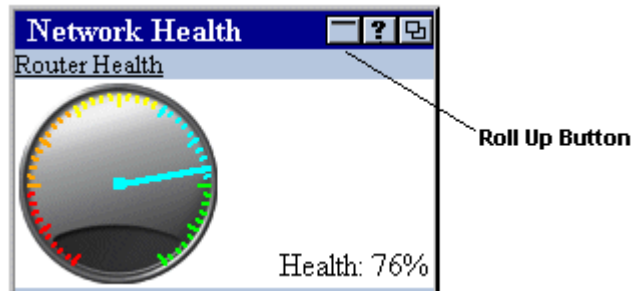
Figure 6-3   Rolldown button
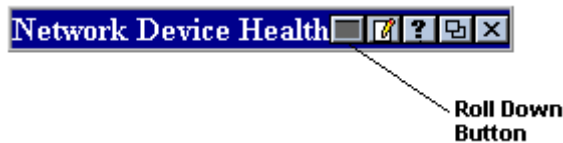
Designing a Custom Look and Feel

SIP provides several ways for you to customize the portal look and feel to suit your company or that of your customers. These are advanced customizations that are made through the direct editing of XML, HTML, and JSP files. Following are the customizations you can make:

- A custom header that can contain your own graphic images.

- A custom footer that matches the header.

- Custom portal 'skins' (portal look and feel).

Customizing the Portal Header and Footer

At the top of the portal page is a header. In the header you can display your own custom logo or that of your customer, as well as a custom background image and text.

Customizing the Header

You will use the supplied default header as the basis of the custom header. After you have defined the custom header, you will override the default header by referencing yours in the customer's portal view file.

1. Go to the directory where the default header is stored:

   Windows: `%SIP_HOME%\webapps\ovportal\jsp\core\`
     UNIX: `/opt/OV/SIP/webapps/ovportal/jsp/core/`

2. Copy and rename the default header file `header.jsp` to a custom header file name (for example, `customheader.jsp`).

3. With an ASCII editor, open the new header file and modify it to change the image files or add text. (The default code for `header.jsp` is shown in Figure 6-4 on page 129.) The following rules apply when modifying the `customheader.jsp` file:

   - The *header* comprises the background image (banner), logo images, and text.

   - The `customheader.jsp` file is an HTML fragment that should NOT have `<HTML><\HTML>` or `<BODY><\BODY>` tags inserted. When editing the HTML, you need to preserve the beginning and end table tags.

- When you make reference to images, use the following path:
  `<img src="/OvSipDocs/C/images/">`

  Make sure you place the image files in the location:
  /OvSipDocs/C/images/

- If you want the banner to show through a logo image, make the logo image transparent.

- The background image (color strip) in the default header is referred to as the *banner*. The default code for the banner is:

`background="/OvSipDocs/Skins/Default/backgrounds/bg-fade.gif"`

This default is only used if the skin is specified to be "[None]."

- Note that `class` equals "`header`" in the *customheader*.jsp file. The `header` class is defined in the cascading style sheet (CSS) files. If you want the header background image (banner) and position, as well as the text color to change based on the selected skin, `class` must equal `header`. But if you want to remove this dependency, you can remove the `class` attribute and value from your *customheader*.jsp file.

Figure 6-4   Default Code for header.jsp

```
<!-- Banner Title bar -->
<table class="header" cellpadding="10" cellspacing="0"
       border="0" width="100%" height="60"
       background="/OvSipDocs/Skins/Default/backgrounds/bg-fade.gif">
   <tr class="header">
     <td class="header" align="left" valign="middle">
       <img src="/OvSipDocs/C/images/custom/generic_net.gif"
            alt="You know it when its Generic!"
            width="141" height="43"/></td>
     <td class="header" align="right" valign="middle">
       <img src="/OvSipDocs/C/images/framework/hplogo.gif"
       alt="Hewlett-Packard" width="140" height="52"/></td></tr>
</table>
```

4. After modifying the text and image references, save and close the *customheader*.jsp file.

5. Next, insert a reference to the new header file into the portal view file of the customer to whom you want to display the customized header. With an ASCII editor, open the portal view file, which is stored in the following location:

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

6. The `PortalView` element can contain the `portalHeader` attribute. (It is not there by default, so you may need to add it.) Type the attribute as shown in the last line of the following example:

Figure 6-5   Example of portalHeader Element Added to a Portal View File

```
<PortalView colorScheme="/OvSipDocs/Skins/Default/Default.css"
          defaultSheetID="Sheet2" refreshRate="3600"
          showDateTime="yes" showUserName="yes"
          userName="Any User"
        portalHeader="customheader.jsp">
```

7. When you are finished, save the portal view file. You can expect to see the change the next time the portal view for this customer is displayed.

8. If you want to change the banner in the header for each of the skins, edit the header class definition in each CSS file that is registered in the `OVPortalConfig.xml` file. CSS files that are registered in the `OVPortalConfig.xml` appear as options in the `Skins` selection list on the `Options` page. (Figure 6-6 on page 131 shows the `header` class definition in the `Default.css` file.) Different banner images can be specified for each CSS file. Place the new background images somewhere under the following directory:

Windows: `%SIP_HOME%\htdocs\C\images\`
UNIX: `/opt/OV/SIP/htdocs/C/images/`

If you change the background image and change the text color, make sure the text color is well coordinated with the colors defined for the `BODY` class in the CSS file. For example, the monochrome CSS file defines a black background for the portal body. If header text is not defined as white, you will not see it.

To perform this type of customization, you need to be proficient in Cascading Style Sheets (CSS) v.1. Visit the W3C site for more information: `http://www.w3.org/Style/CSS`.

Figure 6-6   header Class in default.css

```
/*
* These are resources for the ISP-supplied header HTML.
* These styles may not work depending upon the supplied HTML.
*/
  .header {
    background-image:
                   url("/OvSipDocs/Skins/Default/backgrounds/bg-fade.gif");
    background-position: left top;
    color: black;
         }
```

9. If you want to remove the background image (banner) in the header, you must edit both the *customheader*.jsp and the CSS files. In the *customheader*.jsp file remove the code that specifies the background image. In each CSS file, remove the background-image definition and the background-position definition from the header definition.

Customizing the Footer

If you change the banner color in the header, you should consider doing the same for the footer, so that the colors are coordinated. You can change the color of the footer or place an image in it if you prefer.

The custom footer is defined the same way as the header. In a portal view file, a portalFooter attribute can be added to the PortalView element. If you want to add one, use the same procedure described for portal headers. The class value in the *customfooter*.jsp file would be "footer", and you would just reference the *customfooter*.jsp from the portal view file.

Customizing the Skins

To perform this task, you need to be proficient in Cascading Style Sheets (CSS) v.1. Visit the W3C site for more information: http://www.w3.org/Style/CSS/

Mapping the ColorScheme Element to a CSS File

Before adding or extending existing style sheets it is important to understand the association between the ColorScheme name found in the OVPortalConfig.xml file and the cascading style sheet file.

Here is an example that illustrates the mapping:

On the `Options` page the `Skins` drop-down selection list includes `Seascape`. It appears on the selection list because it was added as a `ColorScheme` element in the `OVPortalConfig.xml` file, as shown below:

```
<ColorSchemes>
   <ColorScheme title="Seascape"
      styleSheet="/OvSipDocs/Skins/Seascape/Seascape.css">
</ColorSchemes>
```

The filename for the Seascape color scheme is `Seascape.css` and is stored in:

Windows: `%SIP_HOME%\htdocs\Skins\Seascape`
UNIX: `/opt/OV/SIP/htdocs/Skins/Seascape`

If you want to add additional items to the "Skins" selection list in the `Options` page, simply add to the existing entries in the OVPortalConfig.xml file and add the corresponding cascading style sheet as described below.

Creating a Cascading Style Sheet

Create a new *mycssfile*.css document and save it in the following directory:

Windows: `%SIP_HOME%\htdocs\Skins\<MyCSS>`
UNIX: `/opt/OV/SIP/htdocs/Skins/<MyCSS>`

You can add new classes to the cascading style sheets, but do not eliminate any of the existing ones. If you do you may inadvertently impact modules that rely on those classes.

Each supplied module defines the default appearance for the module. The cascading style sheets can be changed as a way of overriding the default appearance.

It may be easier to start from an existing file and make changes to it.

When creating a new skin, you may want to create or reference tabs that match your new color scheme. SIP provides a set of layered image files as a starting point for creating tabs. The format of these files is `.psp` (Jasc Paint Shop Pro). They are located in the following directory:

Windows: `%SIP_HOME%\htdocs\C\images\framework\psp\`
UNIX: `/opt/OV/SIP/htdocs/C/images/framework/psp/`

NOTE	If you want to change other graphical images on the portal page (other than those in the banner), you need to use a graphics editor. The interface images are stored in the following directories:

Windows:
```
%SIP_HOME%\htdocs\C\images\framework
%SIP_HOME%\htdocs\C\images\health
%SIP_HOME%\htdocs\C\images\service
```

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

SIP relies heavily upon CSS class names to identify visual regions of the page. In Table 6-3 on page 133 is a description of each region. Refer to the CSS files for additional comments and uses.

Table 6-3   Cascading Style Sheet Classes and Descriptions

Class name	Description Title
BODY	Attributes for the entire page
color	Text color for page
background-color	Background color for page (can be replaced with background-image)
font-style	Style for text on page (normal, italic, oblique)
font-size	Size of text on page (point size, percentage)
font-weight	Thickness of text on page (normal, bold, bolder, lighter)
Below are resources for service provider header HTML. These styles may not work depending upon the supplied HTML:	
.header	Attributes for the header portion of the page
background-image	Image used for banner in header
background-position	Placement for banner in header (top, center, bottom, left, right)
color	Text color for text contained in header

Table 6-3   Cascading Style Sheet Classes and Descriptions (Continued)

Class name	Description Title

`.footer`
  `background-image`
  `background-position`
 	
  `color`
	Attributes for the footer portion of the page
	Image used for banner in footer
	Placement for banner in footer (top, center, bottom, left, right)
	Text color for text contained in footer

Below are resources that control all title and heading styles:	

`.title`
  `color`
  `background-color`
  `font-style`
  `font-weight`
	Primary title for a page or area.
	Text color in the title
	Background color for the title
	Style for text in the title (normal, italic, oblique)
	Thickness of text in the title (normal, bold, bolder, lighter)

`.subtitle`
 	
  `color`
  `background-color`
  `font-style`
  `font-weight`
	Area beneath the primary title that looks like extension of the title (should have same settings as title)
	Text color in the subtitle
	Background color for the subtitle
	Style for text in the subtitle (normal, italic, oblique)
	Thickness of text in the subtitle (normal, bold, bolder, lighter)

`.heading`
 	
  `color`
  `background-color`
  `font-style`
  `font-weight`
	Highlighted area beneath the primary title that has a different but complementary color scheme
	Text color in the heading
	Background color for the heading
	Style for text in the heading (normal, italic, oblique)
	Thickness of text in the heading (normal, bold, bolder, lighter)

`.subheading`
 	
  `color`
  `background-color`
  `font-style`
  `font-weight`
	Area beneath the primary heading that looks like an extension of the heading (should have same settings as heading)
	Text color for the subheading
	Background color for the subheading
	Style for text in the subheading (normal, italic, oblique)
	Thickness of text in the subheading (normal, bold, bolder, lighter)

`.heading A` `color`	Anchor tag within heading Text color for anchor tag in heading

Table 6-3   Cascading Style Sheet Classes and Descriptions (Continued)

Class name	Description Title

`.content`
  `color`
  `background-color`
	Primary data presentation area
	Text color for data presentation area
	Background color for data presentation area

`.content A` `color`	Anchor tag within primary data presentation area Text color for anchor tag in primary data presentation area
Below are resources for the rows of tabs near the top:	
`.tabBar`	The region occupied by the tabs.

`.tabBar.edgeunsel`
  `background-image`
  `background-position`
	Left edge of left-most tab, when unselected
	Image for left edge of left-most tab, when unselected
	Position for image of left edge of left-most tab, when unselected

`.tabBar.edgesel`
  `background-image`
  `background-position`
	Left edge of left-most tab, when selected
	Image for left edge of left-most tab, when selected
	Position for image of left edge of left-most tab, when selected

`.tabBar.unselsel`
 	
  `background-image`
  `background-position`
	Image for overlap between an unselected and selected tab
	Position for image of edge between an unselected and selected tab

`.tabBar.selunsel`
  `background-image`
  `background-position`
	Image for overlap between a selected and unselected tab
	Position for edge between an selected and unselected tab

`.tabBar.unselunsel`
  `background-image`
  `background-position`
	Image for overlap between two unselected tabs
	Position for image of edge between two unselected tabs

`.tabBar.unseledge`
  `background-image`
  `background-position`
	Right edge of right-most tab, when unselected
	Image for right edge of right-most tab, when unselected
	Position for image of right edge of right-most tab, when unselected

`.tabBar.seledge`
  `background-image`
  `background-position`
	Right edge of right-most tab, when selected
	Image for right edge of right-most tab, when selected
	Position for image of right edge of right-most tab, when selected

Table 6-3   **Cascading Style Sheet Classes and Descriptions (Continued)**

Class name	Description Title

`.tabBar.active`
  `color`
  `background-image`
  `background-position`
  `background-repeat`
 	
  `font-style`
  `font-weight`
	Center of selected tab
	Text color for text on the selected tab
	Image for the center of the selected tab
	Position for the image of the center of the selected tab
	Handling of repetitions of the image for the center of the selected tab (repeat-x for tab appearance)
	Style for text in the heading (normal, italic, oblique)
	Thickness of text on selected tab (normal, bold, bolder, lighter)

`.tabBar.inactive`
  `color`
  `background-image`
  `background-position`
 	
  `background-repeat`
 	
  `font-style`
  `font-weight`
	Center of unselected tabs
	Text color for text on the unselected tabs
	Image for the center of the unselected tabs
	Position for the image of the center of the unselected tabs
	Handling of repetitions of the image for the center of the unselected tabs (repeat-x for tab appearance)
	Style for text in the heading (normal, italic, oblique)
	Thickness of text on unselected tabs (normal, bold, bolder, lighter)

`.tabBar A` `text-decoration`	Anchor tag in tab bar Decorations added to the anchor text (underline, overline, line-through, blink, none)
`.tabBar A:hover` `text-decoration`	Hovering mouse over anchor tag in tabbar Decorations added to the anchor text (underline, overline, line-through, blink, none)
Below are resources for the controls on the module title bars:	

`.control`
  `color`
  `background-color`
	Primary controls used on the page
	Text color for primary controls used on page
	Background-color for primary controls used on page

Below are resources that control the tool bar that is usually immediately below the tabBar:	

`.toolbar`
  `color`
  `background-color`
	General toolbar area
	Text color for the general toolbar area
	Background-color for general toolbar area

Table 6-3   **Cascading Style Sheet Classes and Descriptions (Continued)**

Class name	Description Title

`.toolbar .context`
 	
  `width`
  `font-weight`
 	
  `padding-left`
	Display box on the left-hand side of the toolbar that contains the user name and roles drop-down
	Width of display box used for user name and roles
	Thickness of font in the user name area of toolbar (normal, bold, bolder, lighter)
	Amount of spacing on the left hand side of the display box

`.toolbar .status`
 	
  `width`
  `font-size`
  `font-style`
  `font-weight`
 	
  `padding-left`
	Display box in the center of the tool bar that contains the time and date information
	Width of display box used for time and date
	Size of font for time and date (point size, percentage)
	Style for text in the heading (normal, italic, oblique)
	Thickness of font in the user name area of toolbar (normal, bold, bolder, lighter)
	Amount of spacing on the left hand side of the display box

`.toolbar .buttons`
 	
  `color`
  `width`
	Display box on the right-hand side of the toolbar that contains the buttons for options, logout, help, etc.
	Color for text in the buttons in the toolbar
	Width of display box used for buttons

`.toolbar .buttons .button`
 	
  `color`
  `width`
	Display area for the internals of an individual button in the tool bar
	Text color for label in individual button in toolbar
	Width of display area for individual button in toolbar

Below are resources for the entire content area occupied by the modules	

`.modules`
  `border-style`
  `border-color`
  `border-width`
	General settings for modules
	Style for the borders surrounding the modules (solid, double)
	Width of the border surrounding modules

Below are resources for narrow columns. Typically these will be a fixed width and the trailing columns will be as big as they need:	
`.narrow` `width`	Specifications for narrow columns of modules Space to be used by the modules in narrow columns
`.wide`	Specifications for wide columns of modules

Table 6-3   Cascading Style Sheet Classes and Descriptions (Continued)

Class name	Description Title
Below are resources that control the display area for an individual module including the titlebar, heading and content regions. This is used to create a border around the module or to adjust padding between modules:	

`.moduleBox`
  `border-style`
 	
  `border-color`
  `border-width`
	General settings for the outside of each module
	Style for the borders surrounding each module (solid, double)
	Color used for the borders surrounding each module
	Width of the border surrounding each module

Below are resources that control the heading and content areas along (without the title bar):	

`.module`
  `border-style`
 	
  `border-color`
  `border-width`
	General settings for the outside of each module
	Style for the borders surrounding each module (solid, double)
	Color used for the borders surrounding each module
	Width of the border surrounding each module

`.module .title`
  `color`
  `background-color`
  `padding-left`
	Settings for the title area within each module
	Color for text in the title area of each module
	Background color for the title area of each module
	Padding on the left hand side of the title area before the text in the title begins

`.module .title .caption`
 	
  `color`
  `background-color`
  `color`
	Settings for the caption or label on the left-hand side of the module title
	Color for text in the caption within the module title
	Background color for area containing the caption within the module title
	Padding on the left hand side before caption text begins within the module title

`.module .title .controls`
 	
  `color`
  `background-color`
	Settings for the controls on the right on the right hand side of the module title
	Color for text in the control area of the module title
	Background color for the control area of the module title

Table 6-3   Cascading Style Sheet Classes and Descriptions (Continued)

Class name	Description Title

`.module .subtitle`
 	
`  color`
`  background-color`
 	
`  padding-left`
	Area beneath the module title that looks like an extension of the module title (should have same settings as module title)
	Color for text in the caption within the module subtitle
	Background color for area containing the caption within the module subtitle
	Padding on the left hand side before caption text begins within the module subtitle

`.module  .heading`
 	
`  color`
`  background-color`
`  padding-left`
	Highlighted area beneath the module title that contains a label for submodules
	Text color in the module heading
	Background color for the heading
	Padding on the left hand side before the caption text in the module heading

`.module.subheading`
 	
`  color`
`  background-color`
`  padding-left`
	Area beneath the module heading that looks like an extension of the module heading (should have same settings as module heading)
	Text color in the module heading
	Background color for the heading
	Padding on the left hand side before the caption text in the module heading

`.module .content`
`  color`
`  background-color`
`  padding-left`
	Data presentation area within the module
	Text color in the module data presentation area
	Background color for the module data presentation area
	Padding on the left hand side before the module content begins

Below are resources for the edit pages:	

`.editarea`
`  color`
`  background-color`
 	
`  font-size`
	Basic area on which edit controls are presented
	Text color for text presented in the edit area
	Background color for area on which edit controls are presented
	Size of fonts presented on the area where edit controls are presented (point size, percentage)

Table 6-3   Cascading Style Sheet Classes and Descriptions (Continued)

Class name	Description Title

`.editarea .label`
 	
  `color`
  `background-color`
 	
  `font-size`
 	
  `font-weight`
	Highlighted area within edit area that allows for a label to separate different sections of edit controls
	Text color for text presented in the edit area
	Background color for area on which edit controls are presented
	Size of fonts presented on the area where edit controls are presented (point size, percentage)
	Thickness of fonts used in highlighted areas within edit area.

Below are resources for the modules with tables:	

`.oddrow`
  `color`
  `background-color`
	Settings for odd numbered rows in tables (e.g., 1, 3, 5)
	Color for text in odd numbered table rows
	Background color for odd numbered table rows

`.evenrow`
  `color`
  `background-color`
	Settings for even numbered rows in tables (e.g., 1, 3, 5)
	Color for text in even numbered table rows
	Background color for even numbered table rows

Below are the footer resources that control the add drop-down list and anything else at the bottom of the screen:	

`.footer`
 	
  `padding`
  `color`
  `background-color`
	Area below the modules that contains controls for adding modules. Also Service provided footers
	Spacing surrounding the footer area
	Color for text presented in the footer area
	Background color for footer area

Below are the resources that define the Network Health background images for the gauges and detail indicators:	

`.NHGauge .gauge`
 	
  `background-image`
	The high-level gauge in the Network Device Health module.
	Image used to present 3D image of gauge on which values are presented

Table 6-3   Cascading Style Sheet Classes and Descriptions (Continued)

Class name	Description Title
Below are the NHdetail resources that correspond directly with the health ratings defined in the <Rating> tag of the netHealthConfig.xml file:	

`.NHdetail .normal`
`background-image`
 	
`background-position`
	Presentation controls for objects in Network Device Health detail tables that have normal health
	Image used in detail tables to represent objects with normal health
	Position for image within the detail table cell

`.NHdetail .minor`
 	
`background-image`
 	
`background-position`
	Presentation controls for objects in Network Device Health detail tables that have minor severity health
	Image used in detail tables to represent objects with minor severity health
	Position for image within the detail table cell

`.NHdetail .critical`
 	
`background-image`
 	
`background-position`
	Presentation controls for objects in Network Device Health detail tables that have critical severity health
	Image used in detail tables to represent objects with critical severity health
	Position for image within the detail table cell

`.NHdetail .unknown`
 	
`background-image`
 	
`background-position`
	Presentation controls for objects in Network Device Health detail tables that have unknown health
	Image used in detail tables to represent objects with unknown health
	Position for image within the detail table cell

Portal View DTD

The tables below list the elements and attributes you will encounter in the portal view configuration files. The `PortalView.dtd` is located in the same directory as the portal view files:

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

The portal view file contains the following elements:

- `PortalView`
- `Sheet`
- `Column`
- `ModuleInstance`

Table 6-4   **PortalView Element**

Attribute	Description
userName	The user's name as you want it displayed in the salutation on the main portal page. <string> The userName can be overridden by user preference and displayName for the user in the User-Role Model.
refreshRate	The number of seconds to wait before refreshing the user's portal. <positive integer, [0..(2^32)-1]>
colorScheme	Specifies the color scheme to use when displaying the user's portal.
defaultSheetID	Specifies which Tab is active when the portal view is displayed. The value for this attribute should match the id of one of the Sheets defined in the portal view file. Each sheet is assigned a unique id by the management portal when it is created. <string>
showDateTime	Specifies whether or not to display the current time and date on the main portal page. <"yes"|"no">
showUserName	Specifies whether or not to display the user's name on the main portal page. <"yes"|"no"> Can be overridden by user preferences.
portalHeader	This attribute is not present by default. If you want to override the default header, you must add this attribute.

Table 6-4 **PortalView Element (Continued)**

Attribute	Description
portalFooter	This attribute is not present by default. If you want to override the default footer, you must add this attribute.

Table 6-5 **Sheet Element**

Attribute	Description
title	Specifies the name of the tab. <string>
id	A unique identifier for a sheet in the portal view file. The defaultSheetID attribute for the PortalView element will be set to the id for one of the sheets defined in the portal view file. The id string MUST begin with an alpha character, not a numeric character. <string>

Table 6-6 **Column Element**

Attribute	Description
width	A required attribute that specifies the width of the column to be placed on the sheet. Acceptable values are (narrow | NARROW | wide | WIDE).

Table 6-7 **ModuleInstance Element**

Attribute	Description
title	Specifies the name of the module. This value is used as the module name in the portal view. <string>
id	A unique identifier for a ModuleInstance. This identifier is used by SIP to differentiate between instances of the same class within the portal view XML file. id is defined as an ID-tokenized attribute, meaning that an XML file using this attribute for this ModuleInstance element must specify unique text for each element. The id string MUST begin with an alpha character, not a numeric character. <string>

Table 6-7   **ModuleInstance Element (Continued)**

Attribute	Description
classid	The unique identifier for a module class. This string is in the form: classid="com.hp.ov.portal.modules.alarms" This string is used by SIP to help identify which module to load, and to generate instances of a module class. The classid for a module is defined in the module registration file *OVModuleRegistration*.xml *Windows:* %SIP_HOME%\registration\ *UNIX:* /etc/opt/OV/SIP/registration/
help	Specifies the URL to the help content for this module instance. This attribute allows you to override the default help URL defined in the module registration file. If you place your help topic somewhere under the /OvSipDocs directory, your topic will be displayed in the same decorative window in which SIP help topics are displayed. The recommended format is: help="/OvSipDocs/C/help/<mod_directory>/<*topic*>.html"
rollupState	Specifies whether the module is currently rolled "up" or rolled "down." The only thing visible on a rolled up module is its title bar and, if applicable, its instance headers. A module that is rolled down displays everything. <"up"|"down">

7   Developing Portal Content

Understanding the Modules That Are Supplied With SIP

Service Information Portal provides several out-of-the-box modules. These are:

- Modules that present information from HP OpenView management products, such as Network Node Manager (NNM), OpenView Operations (OVO) and Service Navigator (OVSN), OpenView Operations for Windows, OpenView Internet Services (OVIS), OpenView Service Desk (OVSD), OpenView Performance Insight (OVPI), OpenView Reporter (OVR), and OpenView Performance Manager (OVPM).

- Modules that present general information such as links to URLs and messages to your customers.

- Third-party integrations that are based on the SIP Generic module. (These are unsupported contributed modules.)

- The SIP Generic module, which you can use to develop modules that integrate your own application and data.

For a listing of all supplied modules, see Table 3-2 on page 36, Table 3-3 on page 40, and Table 3-4 on page 41.

Modules that Present Information from HP OpenView Management Products

Before you can display data through these modules, you must establish a connection between SIP and the management products. For more information, see Chapter 4, "Connecting SIP to Your Management Products," on page 67.

For detailed information on configuring and using these modules, see the separate books located under the following directory:

Windows: `%SIP_HOME%\htdocs\C\manuals\`
UNIX: `/opt/OV/SIP/htdocs/C/manuals/`

To create your own online help topics for the supplied modules, see "Adding and Customizing Module Help Topics" on page 157.

Modules That Display Bookmarks and Messages

Two supplied modules deliver content that is not related to management data: the Message Board module and Bookmarks module.

- **Message Board Module**: A way of getting important information to your customers through the portal interface. Refer to "Sending Messages to Your Customers through the Message Board Module" on page 149 in the current chapter.

- **Bookmarks Module**: A way of integrating website links into the portal interface. Refer to "Providing Links to Other Websites Via the Bookmarks Module" on page 153 in the current chapter.

To create your own online help topics for the supplied modules, see "Adding and Customizing Module Help Topics" on page 157.

Contributed Integrations Based on the Generic Module

Several unsupported contributed integrations of other HP and third-party products have been developed and provided with SIP. Note that these are contributed modules and are not supported. They are located under the following directory:

Windows: `%SIP_HOME%\integrations\`
UNIX: `/opt/OV/SIP/integrations/`

See the `Readme` files in the directories under the `integrations` directory for instructions on registering and configuring these modules. Until they are explicitly registered, these modules are not available through the portal.

To create your own online help topics for these modules, see "Adding and Customizing Module Help Topics" on page 157.

The Generic Module for Developing Your Own Modules

In addition to presenting management data from OpenView products, you can integrate other applications and data through use of the Generic module.

The **Generic module** provides a non-programmatic way to easily and quickly incorporate into SIP your existing web applications, reports, and other data. You can create your own full-featured modules that can be added and customized through the portal interface, extending the functionality of SIP without writing portal-specific Java code.

Documentation on using the Generic module to develop your own modules is covered in a later section of this chapter.

Sending Messages to Your Customers through the Message Board Module

The Message Board provides a way of getting important information to your customers through the portal.

Developing Messages to Send Your Customers through the Portal

Messages are customizable through the Message Board Administration page. From this page you can create and manage the messages that appear in the `Available Messages` selection list on the Message Board Edit page. You can:

- Create new messages that include information such as the status of problems in the managed environment or new services that you want to offer your customers.

- Change the content of existing messages.

- Delete from the Message Board edit page (accessed from the module title bar) those messages that no longer should be available for display in customized portals.

TIP   In a portal view, if you reference a message file that contains no content (the message file length (size) is zero), the message will not be displayed. You can use this feature to prepare your portal views for a future message that you may want to communicate with some urgency. For example, if you assign to your portal views a message called "urgent" but leave the content blank, you can quickly disseminate information through this message in an emergency situation. Just be sure to remove the content of the message file after the emergency and not the file itself. If you delete a file that is assigned to a portal view, the portal will display the message "Data currently unavailable."

Accessing the Message Board Administration Page

1. Log in as a user who has access to a special SIP Administrator role. For more information, "Understanding Special SIP Administrator Roles" on page 79.

2. Switch to the SIP Administrator role, if it is not already displayed.

3. Click the Message Board tab.

Creating Messages

1. On the Message Board tab, click New.

2. In the File Name field, type a message file name that is unique, contains no spaces, and contains no non-alpha-numeric special characters. Also, the file name cannot begin with a forward slash (/) or with two periods followed by a forward slash (../).

   This file name is the HTML file that will contain the message content. Message files are stored in the following directory:

   Windows: %SIP_HOME%\conf\share\modules\messageboard\
   UNIX: /opt/OV/SIP/conf/share/modules/messageboard/

3. Type the name that you want to appear in the portal as the message title.

4. Type the message content.

5. Create the message and return to the main Message Board page by clicking [OK].

Changing the Content of a Message

If the message that you are changing is presented to multiple user roles, all of them will see the changes you make to the content.

Also, once you create a message, you cannot modify its HTML file name. Instead you must delete the file and create a new one with the new file name.

1. On the Message Board tab, select a message and click [Modify].

2. Modify the message name and content, as desired.

3. Save modifications and return to the main Message Board page by clicking [OK].

Deleting Messages

When you delete a message, you are deleting it from the `Available Messages` selection list only. (This list appears for a portal view when you click the edit button on the Message Board title bar.) If you perform this delete action while the message is still assigned to a portal view, it will continue to appear in the portal view until you manually delete it from the portal view through the `Message Board Edit` page.

1. On the `Message Board` tab, select a message and click `[Delete]`.

2. When prompted, confirm the delete action.

3. Log in to each portal view that displays the message and remove the message from the Displayed Messages list accessed from the edit button on the module title bar.

4. Delete the message from the file system, if desired. Messages are stored in the following directory:

   Windows: `%SIP_HOME%\conf\share\modules\messageboard\`
   UNIX: `/opt/OV/SIP/conf/share/modules/messageboard/`

Even after a message is deleted from both the `Available Messages` selection list and the portal views, the HTML file that contains the message content remains on the file system. You can delete it if you like. If you do not, you will not be able to reuse the message's file name.

Displaying Messages in Portal Views

After you have created messages through the `SIP Administration Pages`, they appear on the `Available Messages` selection list. This list appears when you are in a portal view and click the edit button on the Message Board title bar.

Choosing Messages to Be Displayed In a Portal View

1. Log in as a user who has access to the role you want to display messages to, and switch to the role.

2. Navigate to the Message Board module, and click the edit button on the module title bar.

3. On the `Message Board - Edit` page, select a message in the `Available Messages` field, and click `[Add]`.

4. Repeat the previous step until the `Displayed Messages` field contains all the messages you want to display to users in the current role.

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

Removing Messages from a Portal View

1. Log in as a user who has access to the role that displays the message you want to remove.

2. Navigate to the Message Board module, and click the edit button on the module title bar.

3. On the `Message Board - Edit` page, select a message in the `Displayed Messages` field, and click `[Remove]`.

4. Repeat the previous step until the `Displayed Messages` field contains only the messages you want to display to this customer.

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

Changing the Display Order of Messages in the Portal View

1. Log in as a user who has access to the role that displays the message you want to change, and switch to the role.

2. Navigate to the Message Board module, and click the edit button on the module title bar.

3. On the `Message Board - Edit` page, reorder the `Displayed Messages` by selecting a message and clicking `[Up]` or `[Down]`.

4. Repeat step 3 until the `Displayed Messages` field lists the messages in the order you prefer.

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

Providing Links to Other Websites Via the Bookmarks Module

The Bookmarks Module gives you a way to provide in your customized portals lists of links to useful web sites.

Developing Groups of Shared Bookmarks

A convenient way to manage the lists of bookmarks is to create groups of shared bookmarks, which can be assigned to multiple portal views. When you change the bookmarks, they are automatically updated in all portals that display them.

Through the SIP Administration Pages you can create, modify, and delete shared bookmarks. Through the edit button on the Bookmarks module title bar, you can assign or remove bookmarks from individual portal views.

Accessing the Bookmarks Administration Page

1. Log in as a user who has access to a special SIP Administrator role. For more information, "Understanding Special SIP Administrator Roles" on page 79.

2. Switch to the `SIP Administrator` role, if it is not already displayed.

3. Click the `Bookmarks` tab.

Creating a Shared Bookmarks Group

1. On the `Shared Bookmark - Administration` page, click `Add`.

2. In the `Group Name` field, type a name that is unique and contains no spaces.

3. In the `Title` field, type the name that you want to appear in the portal as the shared bookmarks group.

4. Define a bookmark by supplying the following entries:

   • `Bookmark Name` - The name of the link as it is displayed to users through their portal views.

- `URL` - The URL link to the bookmark. You must include http:// in the URL. For example, use http://hp.openview.com instead of hp.openview.com. See table

- `Window Name (optional)` - The name of the new window to create (or reuse) when opening the URL so that the current portal window is not replaced. The value should contain no spaces.

5. Add the newly defined bookmark to the current shared group by clicking [Add To List].

6. Repeat steps 4 and 5 until you have added all bookmarks for this shared group.

7. When finished, return to the main `Bookmarks Administration` page by clicking [OK].

Modifying a Group of Shared Bookmarks

1. On the `Shared Bookmark - Administration` page, select a shared group and click [Modify].

2. To add a bookmark, define the `Bookmark Name`, `URL` and `Window Name` (optional) and then click [Add To List].

3. To modify an existing bookmark, select it from the `Current Bookmarks` list, and click [Modify]. Edit the bookmark definition (Bookmark Name, URL, and Window Name), and then click [Add To List].

4. To delete a bookmark from the shared group, select it from the `Current Bookmarks` list, and click [Delete].

5. To reorder the list of bookmarks in the group, select one from the `Current Bookmarks` list, and click [Up] or [Down].

6. When finished, return to the main `Bookmarks Administration` page by clicking [OK].

Deleting a Group of Shared Bookmarks

It is recommended that before you delete a group of shared bookmarks, you remove it from each of the portal views to which it is assigned.

When you delete a shared group, you are deleting it from the Shared Bookmarks list only. This list is displayed through a portal view by clicking the edit button on the Bookmarks module title bar. (The Shared

Bookmarks list is stored in the sharedbookmarks.xml. For more information about the module XML files, in Appendix F refer to "Creating Groups of Shared Bookmarks Through Direct Editing of XML" on page 345.)

If you delete a shared bookmark group before removing it from the portals that display it, in its place the Bookmark modules will display "Data Currently Unavailable."

To remove it from portal views, log in to each portal that displays it, navigate to the Bookmarks module, and click the edit button on the module title bar. From the Current Bookmarks list, select the shared group and click [Remove].

Table 7-1   **Default Port and Protocol Settings for Bookmarks**

Server Product	Protocol	Port	Configuration
Various	http	80	Server, port, and protocol configurable through the URL specified through the Bookmarks tab of the SIP Administration Pages or the Bookmarks module edit page accessed from the module title bar.
	https	443	

Displaying Bookmarks in a Portal View

If you want to display groups of shared bookmarks through a portal view, you must first create them through the SIP Administration Pages. (See "Creating a Shared Bookmarks Group" on page 153.) After they are created, they appear on the Available Shared Groups selection list that is displayed when you are in a portal view and click the edit button on the Bookmarks title bar.

NOTE   Bookmark groups (not to be confused with shared bookmark groups) can only be edited and configured through direct editing of the XML. For instructions, see "Grouping Entries" on page 349.

Choosing Bookmarks to Be Displayed In a Portal View

1. Log in as a user who has access to the role you want to display bookmarks to, and switch to the role.

2. Navigate to the Bookmarks module, and click the edit button on the module title bar.

3. On the `Bookmarks - Edit` page, add a bookmark by clicking [Add Bookmark] and then defining it.

4. Add a shared bookmark by clicking `[Add Shared Group]` and selecting it from the list of `Available Shared Groups`.

5. Repeat the previous step until the `Current Bookmarks` list contains all the bookmarks you want to display to users in the current role.

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

Removing Bookmarks from a Portal View

1. Log in as a user who has access to the role that displays the bookmark you want to remove.

2. Navigate to the Bookmark module, and click the edit button on the module title bar.

3. On the `Bookmarks - Edit` page, select a bookmark from the `Current Bookmarks` list, and click `[Remove]`.

4. Repeat the previous step until the `Current Bookmarks` list contains the bookmarks you want to display through the current role.

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

Changing the Display Order of Bookmarks in the Portal View

1. Log in as a user who has access to the role that displays the bookmarks you want to reorder, and switch to the role.

2. Navigate to the Bookmarks module, and click the edit button on the module title bar.

3. On the `Bookmarks - Edit` page, reorder the `Current Bookmarks` by selecting a bookmark and clicking `[Up]` or `[Down]`.

4. Repeat step 3 until the list is arranged in the order you prefer.

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

Adding and Customizing Module Help Topics

SIP provides the capability to add your own help topic for any of the supplied modules, including Generic-based modules. You can provide a help topic for the default module, and also override it for an individual portal view.

When the user clicks the "?" button on the module title bar, the module will display the specified HTML help file.

Adding/Customizing a Default Help Topic

All modules that are supplied with SIP come with a default help topic, which is specified in the module registration file. If you want to customize the help topic or add a new one (as you may for your own Generic-based modules), you need to write the help topic, assign it as the default help topic (or a new topic that overrides the default help topic), and verify that the topic displays correctly.

Writing a Help Topic

1. Copy and rename any SIP help topic file. The new file can have any name that makes sense to you. Help files are stored in the directories under the following directory:

   Windows: `%SIP_HOME%\htdocs\C\help\`
   UNIX: `/opt/OV/SIP/htdocs/C/help/`

2. Write your help file content. Feel free to present a help page using the most appropriate style for the module, but follow these important guidelines:

   • Follow the HTML format that you find in the other files in the `help` directory.

   • Use HTML syntax that is supported by all web browsers.

   • Help is written for the module as a whole, not for individual submodules.

   • Use only ASCII or UTF-8 characters.

3. Save the new HTML help topic file and place it in the correct location:

Windows: `%SIP_HOME%\htdocs\C\help\`
UNIX: `/opt/OV/SIP/htdocs/C/help/`

Assigning a Topic as the Default Help Topic for a Module

1. Edit the module registration file located in the following directory:

   Windows: `%SIP_HOME%\registration`
   UNIX: `/opt/OV/SIP/registration`

2. Define/redefine the new help file path and name in the help attribute. The `help` attribute must be a URL and can point at documentation from the local system. The syntax should be:

   `help="/OvSipDocs/C/help/<mod_directory>/<topic.html>"`

   To see a sample module registration file, refer to Figure 7-26 on page 190.

3. Save the registration file.

4. Restart the servlet engine. For instructions, see "Restarting the Servlet Engine" on page 291.

Verifying that a Help Topic Displays

1. In a browser, log in as the appropriate SIP user.

2. If you have not already done so, add the module to the portal view.

3. Click `[?]` on the module title bar to ensure that the desired behavior is established.

Overriding the Default Help Topic

For a module instance within a portal view file, you can override the default help topic that is defined in the module registration file.

1. Write the new help topic and place it in the correct location, as described in "Writing a Help Topic" on page 157.

2. Using an ASCII text editor, open the portal view XML file that contains the module instance.

3. Navigate to the module instance. Insert the `help` attribute and specify the help value.

The `help` attribute must be a URL and can point at documentation from the local system. For example:

```
<ModuleInstance id="myModule
        title="MyModule"
        help="/OvSipDocs/C/help/MyDirectory/MyHelp.html">
```

4. Save and close the XML file.

5. Validate the XML, as described in "Validating XML Files" on page 313.

6. Verify that the topic display correctly, as described in "Verifying that a Help Topic Displays" on page 158.

Developing Modules that Integrate Your Own Applications and Data

In addition to presenting management data from OpenView products, you can integrate other applications and data through use of the Generic module. It gives you a way to provide additional information to your customers beyond what the supplied modules offer.

The **Generic module** is a non-programmatic way to easily and quickly incorporate into SIP your existing web applications, reports, and other data. You can create your own full-featured modules that can be added and customized through the portal interface, extending the functionality of SIP without writing portal-specific Java code.

Using the Generic module based on a simple specification in an XML file, you can create a module that is configurable through a user interface.

The Generic module also provides limited proxying capabilities for the URLs. This way, you can integrate into portal web sites that are behind a firewall and are not accessible to your end customer.

The Generic module offers you the following capabilities:

- **Display content from a specified URL**. For example, if you have an HTML file on a different server or a CGI program or other web application, you can display the contents of the URL by referencing the URL in a portal view file.

- **Display output from executable commands**. For example, if you want to run a command, such as `ping` or a command that generates results from a database, and display the output of the command in the portal, you can enter the command in a portal view file.

- **Display HTML from an external file**. For example, if you have a report in an HTML file on the local machine or a tool that generates reports in the form of text, you can display those reports through the portal by referencing the HTML file in a portal view file.

- **Display Content from XML that has been transformed using XSLT**. For example, if you have data in XML format that is generated from a reporter engine or a CGI program and that you typically present using a XSL transformation, SIP can transform this data for you and display it in the portal.

- **Display embedded HTML**. For example, if you have a tool that generates reports in the form of a GIF image, you can display the report through SIP by embedding the HTML in a portal view file.

For examples of the Generic module, log into SIP as `admin` user and switch to the `Integration Examples` role or `Demo` role. All modules in these two roles are based on the Generic module. Another example is the "Service Desk for Specialists" module in the `Help Desk` tab of the `LiveDemo` role.

TIP   Generic modules can be added through the portal interface and customized through an Edit GUI. You can use the Generic module to create modules that have all of the functionality of a supplied, full-featured module.

Process of Creating Integrated Modules

The diagram below summarizes the process of creating integrated SIP modules based on the Generic module.

Figure 7-1   **Process of Creating Modules Based on the Generic Module**
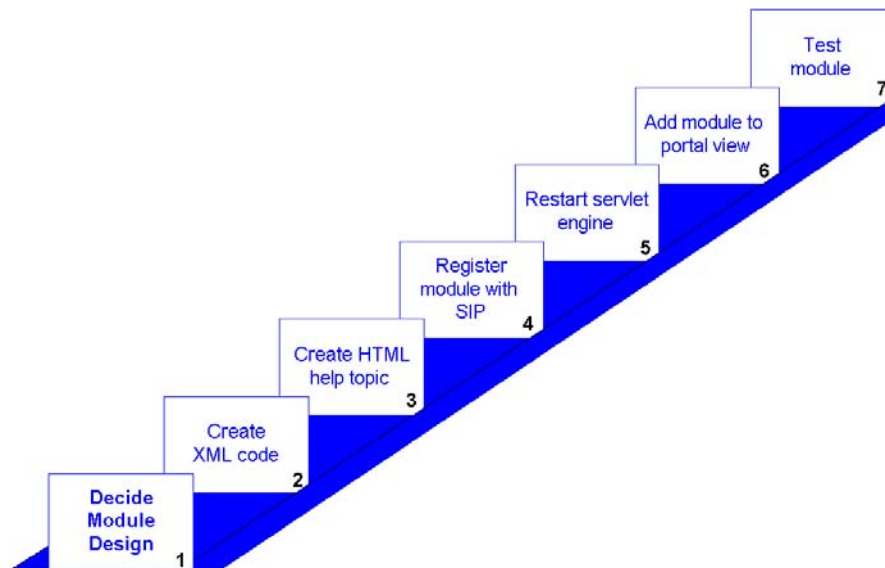
You can easily understand the basic process of creating integrated modules by following a short tutorial provided in the next section.

Going through the Process: A Tutorial

You can easily understand the basic process of creating integrated modules by following this short tutorial. You are instructed to take a supplied, sample module that is based on the Generic module (in lieu of creating XML code, Step 2), register it with SIP (Step 4), restart the servlet engine (Step 5), and add it to a portal view (Step 6). The sections following the tutorial describe the steps in detail.

Create the XML

For this tour, you will use an existing default module file. Provided with SIP are several sample integrations that are based on the Generic module. Instead of developing XML for this exercise, you will use one of the sample integrations. (Note that default module XML must be placed in a specific location.)

The default module file contains a snippet of XML that makes up a complete Generic-based module. This code is added to a portal view file when you add a module through the portal interface.

Register an Integration Module with SIP

The registration file defines such information as a unique ID for the module, a reference to the servlet, the help file associated with the module, and the capabilities of the module.

1. Go to the following directory where the sample integration code and sample registration file are stored:

   Windows: `%SIP_HOME%\integrations\Yahoo_Headlines`
   UNIX: `/opt/OV/SIP/integrations/Yahoo_Headlines`

2. Copy the files `OVDefaultYahooHeadlines.xml` and `OVRegYahooHeadlines.xml` into the following directories, which are required locations:

   - `OVDefaultYahooHeadlines.xml`

     Windows: `%SIP_HOME%\registration\defaults`
     UNIX: `/opt/OV/SIP/registration/defaults`

   - `OVRegYahooHeadlines.xml`

Windows: `%SIP_HOME%\registration\`
UNIX: `/opt/OV/SIP/registration/`

Restart the Servlet Engine

After adding or changing a module registration file, you must stop and restart the servlet engine before the changes take effect in SIP. For instructions on doing so, see "Restarting the Servlet Engine" on page 291.

Add the Module to a Portal View

1. Start SIP by opening a browser and entering the following:
   http://<yourhostname.com>/ovportal

2. Log in as **admin**.

3. From the Role selection list in the portal banner, select the role `Create A View`. Go to the [Add] button at the bottom of the wide tab column.

4. Select `Yahoo Headlines` and click [Add].

Resources for Creating Generic-Based Modules

Once you know the process for creating Generic-based modules, all you need is the supplied sample code and an understanding of the DTD. Below is a listing of resources and where you can find these.

Sample registration file

- *Windows:*

`%SIP_HOME%\integrations\Yahoo_Headlines\OVRegYahooHeadlines.xml`

- *UNIX:*

`/opt/OV/SIP/integrations/Yahoo_Headlines/OVRegYahooHeadlines.xml`

Sample Generic-based Modules: OpenView Integration

Service Desk is a registered module that does single sign-on and proxying. It is located in:

Windows: `%SIP_HOME%\registration\`
UNIX: `/opt/OV/SIP/registration/`

Sample Generic-based Modules: Third-Party Integrations

Concord, Keynote, NNM Commands, NNM Reports, Opticom,
OV Service Desk, OV Reporter, Webtrends, and Yahoo Headlines:

Windows: `%SIP_HOME%\integrations\`
UNIX: `/opt/OV/SIP/integrations/`

Location of Default Module files

Windows: `%SIP_HOME%\registration\defaults\`
UNIX: `/opt/OV/SIP/registration/defaults/`

Sample Portal View files

`cannedDemo.xml` and `integration.xml` located here:

Windows: `%SIP_HOME%\conf\share\views\samples\`
UNIX: `/opt/OV/SIP/conf/share/views/samples/`

DTD and Element Definitions

`OVGeneric.dtd` located here:

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

`OVModuleRegistration.dtd` located in the following directory:

Windows: `%SIP_HOME%\registration\`
UNIX: `/opt/OV/SIP/registration/`

Rules for Editing XML Files

Designing the Functionality of Your Module

Before you create a Generic-based module, you need to decide on the
functionality you want to build into it. Answer for yourself the following
questions:

- Do you want to provide for editing of the module configuration from
  the portal interface?

For example, you can provide an editing GUI with a choice list of module parameters. You can allow a person with `ViewAdmin` edit permissions to make these types of module configurations. For more information on edit permissions, see "Understanding Basic Roles" on page 78.

- If you provide an edit GUI, what parameters will you offer?

  For example, parameters can be hosts, various reports, different commands, and for the different commands ability to have different parameters.

- Will the parameters be selected from a options list, or will they be entered into a text field?

  For example, you can create a selection list of commands, or allow the text entry of a command.

- Will your module have multiple submodules?

- Because editing GUIs apply to the entire module and not individual submodules, how will you design the editing interface appropriately for all submodules?

- Do you want to develop an online help topic for your module?

  If you do, keep in mind that, like edit GUIs, the help topic applies to the entire module and not individual submodules.

NOTE To learn more about the kinds of choices you have when designing a Generic-based module, refer to the examples in "Displaying the Output from a Command" on page 167. Four examples illustrate the output of the "ping" command displayed through the portal; but each example uses a different approach.

Creating XML for a Generic-Based Module

The process of creating the XML code for a Generic-based module can be summarized in the following steps:

1. Find an example of a Generic-based module that is similar to the one you want to create. Make sure you copy the entire `Generic` element from the module's default XML file.

NOTE	For the location of example modules, see "Sample Generic-based Modules: Third-Party Integrations" on page 164 and "Sample Portal View files" on page 164. You can also create a module from scratch. For detailed information on the XML elements and attributes, see the documentation that begins on page 193.

2. Copy and rename the default XML file that contains the example. Place the file in the defaults directory where the default module code for each registered module is stored:

   Windows: `%SIP_HOME%\registration\defaults\`
   UNIX: `/opt/OV/SIP/registration/defaults/`

3. Using a text editor, open the default file and modify the XML code.

   For information and sample code that is specific to each type of submodule, see:

   - "Displaying the Output from a Command" on page 167

   - "Displaying HTML from an External File" on page 170

   - "Displaying Embedded HTML" on page 173

   - "Displaying the Contents of a URL" on page 175

   - "Transforming XML Into a Displayable Format Using XSLT" on page 177

   For additional functionality you can add to your module, see:

   - "Proxying Capabilities" on page 178

   - "Generic Module Single Sign-on" on page 178

   - "Giving Your Module Access to SIP Data Through Variable Substitution" on page 180

   - "Adding an Edit GUI to Your Module" on page 186

   - "Adding Status Text, Graphics, and Links to a Submodule Title" on page 187

4. Save the file to the following directory:

   Windows: `%SIP_HOME%\registration\defaults\`
   UNIX: `/opt/OV/SIP/registration/defaults/`

Displaying the Output from a Command

Through SIP, you can display the output from program execution. In fact, you can capture anything generated to stdout. For example, if you want to run a command, such as ping or a command that generates results from a database, and display the output of the command in a portal, you can achieve this through the Generic-based module.

To display the command output, you will define the Command element in a Generic element. See the examples in Figure 7-2 through Figure 7-11 and the element definitions in Table 7-7 on page 199.

CAUTION Be careful when presenting the results of a command. Unless configured differently, a command is executed as:

Windows: the local "System" account
HP-UX: www/www
Solaris: nobody/nogroup

Make sure that only knowledgeable personnel implement this feature.

Figure 7-2 **Example Ping Command**

```
<Generic>
  <Submodule>
    <TitleBar title="Ping Command"/>
    <Command commandLine="ping.exe -n 3 localhost"
     expires="5" stripHtmlHeader="no" type="text/plain"/>
  </Submodule>
</Generic>
```

Figure 7-3 **Example Ping Command: Three Machines**

```
<Generic>
  <Submodule>
    <TitleBar title="Ping Local Host"/>
    <Command commandLine="ping.exe -n 3 localhost"
      expires="5" stripHtmlHeader="no" type="text/plain">
    </Command>
  </Submodule>
  <Submodule>
    <TitleBar title="Ping weminuche"/>
    <Command commandLine="ping.exe -n 3 weminuche.cnd.hp.com"
```

```
        expires="5" stripHtmlHeader="no" type="text/plain">
      </Command>
    </Submodule>
    <Submodule>
      <TitleBar title="Ping fcbeyond"/>
      <Command commandLine="ping.exe -n 3 fcbeyond.cnd.hp.com"
        expires="5" stripHtmlHeader="no" type="text/plain">
      </Command>
    </Submodule>
  </Generic>
```

Figure 7-4   **Example Ping Command with Editing GUI: Option Parameter**

```
<Generic>
  <Submodule>
    <TitleBar title="Ping Command"/>
    <Command commandLine="ping.exe -n 3 $HOSTNAME"
      expires="5" stripHtmlHeader="no" type="text/plain">
      <OptionParm name="HOSTNAME"
        prompt="Choose Ping Target" value="Local Host">
          <Option name="Local Host" value="localhost"/>
          <Option name="weminuche"
                  value="weminuche.cnd.hp.com"/>
          <Option name="fcbeyond"
                  value="fcbeyond.cnd.hp.com"/>
      </OptionParm>
    </Command>
  </Submodule>
</Generic>
```

CAUTION   Be aware that Figure 7-4 on page 168 and Figure 7-5 below are example only; for security reasons, you may not want to offer a text parameter or options parameters for commands.)

Figure 7-5   **Example Ping Command with Editing GUI: Text Parameter**

```
<Generic>
  <Submodule>
    <TitleBar title="Ping Command"/>
      <Command commandLine="ping.exe -n 3 $HOSTNAME"
        expires="5" stripHtmlHeader="no" type="text/plain">
      <TextParm name="HOSTNAME" prompt="Ping Target"
```

```
      value="localhost"/>
        </Command>
   </Submodule>
</Generic>
```

NOTE   To test that the functionality works in the portal, reference the following:

- "Registering a Module with SIP" on page 188

- "Restarting the Servlet Engine" on page 191

- "Adding the Module to a Portal View" on page 191

- "Testing a New Module" on page 192

Displaying HTML from an External File

Through SIP, you can display HTML from an external file. For example, if you have a tool that generates reports in the form of a GIF image or text, you can display that report through the portal by referencing the HTML file.

To display HTML from an external file, you will define the `File` element in a Generic element. See the examples in Figure 7-6 on page 171 and the element definitions in Table 7-8 on page 202.

Rules for External HTML

- The file must be on your local machine or network. It cannot require http access. If you want to display files on a remote web server, then refer to "Displaying the Contents of a URL" on page 175.

- The file can be HTML or plain text. The `type` should be set to "`text/html`" when the content is HTML. The `type` should be set to "`text/plain`" when the content is plain text.

- The HTML should be well formed.

- The contents of an external HTML file should not include "header" tags such as `<HTML></HTML>`, `<HEAD></HEAD>`, or `<BODY></BODY>`. The portal will automatically provide these tags. If your HTML files include these tags, you can use `stripHTMLHeader`. For more information about `stripHTMLHeader`, refer to Table 7-7 on page 199.

- If you have style sheets in a file, your imported style sheets may be ignored. If you embed content from a URL that defines a cascading style sheet, it will override the SIP cascading style sheet.

Enabling SIP to Access Remote Files

The user that SIP is running as must have access to the remote files. By default, the Tomcat process is configured to run under the user "www" (on HP-UX) and "nobody" (on Solaris). On Windows, Tomcat is configured to run as the "system" account.

A potential problem is that the user may have a different user id on the SIP server than it does on the remote system. For example, on the SIP system, user "www" has user id 101. On the remote system with the `/conf/share` directory, "www" has user id 30. When the id is passed in the authentication check, the authentication will fail.

On UNIX, the solution is to make sure that "www" or "nobody" has the same user id on both systems. Another possibility is running Tomcat as some other new or existing user, provided that both systems agree on the user_name/user_id mapping.

On Windows, by default, the Tomcat service is configured to run as the "system" account. When SIP tries to access shared resources on Windows, it only has the default, local system account privileges, and may not have access to shared drives on other machines. In this situation, configure the Tomcat service to be executed as a user who has access to the remote directory. The user should have "read" and "change" privileges to the remote directory.

In the Windows Services window, select the Tomcat service and view the properties. Select the Log On tab and specify the username and password of the user that Tomcat should run under. Stop and restart the service, and SIP will have the privileges that are granted to the given user.

Figure 7-6   **Example: Displaying HTML from an External File from cannedDemo.xml**

```
<Generic>
  <Submodule>
    <TitleBar titleAnchorText="Router Health"
      titleAnchorURL="/OvSipDocs/C/demo/router_details.html"/>
    <File displayFileInfo="no"
      fileName="$SIP_HOME_DIRhtdocs/C/demo/router_health.html"
     stripHtmlHeader="yes" type="text/html"/>
  </Submodule>
</Generic>
```

Figure 7-7   **Example of Displaying External HTML Files With Editing GUI: Option Parameters**

```
<Generic>
  <Submodule>
    <TitleBar titleAnchorText="Employee Files"/>
    <File displayFileInfo="no"
      fileName="$SIP_HOME_DIRhtdocs/C/demo/$EmpFiles"
      stripHtmlHeader="yes" type="text/html">
    <OptionParm name="EmpFiles"
      prompt="Choose Employee File" value="John">
     <Option name="John" value="john.html"/>
     <Option name="Jane" value="jane.html"/>
```

```
        </OptionParm>
      </File>
    </Submodule>
</Generic>
```

NOTE   To test that the functionality works in the portal, reference the following:

- "Registering a Module with SIP" on page 188
- "Restarting the Servlet Engine" on page 191
- "Adding the Module to a Portal View" on page 191
- "Testing a New Module" on page 192

Displaying Embedded HTML

Through SIP, you can embed a short HTML snippet in the portal view file. To display embedded HTML, you will define the `EmbeddedHtml` element in a Generic element. See the examples in Figure 7-8 on page 173 and the element definitions in Table 7-10 on page 210.

Rules for Embedded HTML

- Embedded HTML should be well formed.

- Embedded HTML should be brief; if the HTML content is lengthy, refer to "Displaying HTML from an External File" on page 170.

- The contents of embedded HTML should not include "header" tags such as `<HTML></HTML>`, `<HEAD></HEAD>`, or `<BODY></BODY>`. The portal will automatically provide these tags.

- The value you enter for embedded HTML may be any string of text not containing a less-than sign (<) or quotation marks ("), or similar characters. These characters may be inserted using the usual entity references (`<`, and `"`) or by their Unicode values using character references. All raw ampersands (`&`), those that do not begin a character or entity reference, must also be escaped as `&`. Embedded HTML must not contain: "]]>"

Figure 7-8   **Example of Displaying Embedded HTML from cannedDemo.xml**

```
<Generic>
  <Submodule>
    <TitleBar title="Email Service"/>
    <EmbeddedHtml data="&lt;img height=79 width-239
     src=&quot;/OvSipDocs/C/demo/services/email-card.gif&quot;
     alt=&quot;/Email Service Card - Minor&quot;&gt;"/>
  </Submodule>
</Generic>
```

Figure 7-9   **Example of File and Embedded HTML**

```
<Generic>
  <Submodule>
    <TitleBar title="Team Photo"/>
      <File displayFileInfo="no"
          fileName="$SIP_HOME_DIRhtdocs/C/demo/teamphoto.html"
          type="text/html"/>
        <EmbeddedHtml data="&lt;br&gt;&lt;i&gt;Standing
```

```
              (from Left): Jane, Joe, Janet, Jack. Sitting
              (from left): Bob, Bill, Betty, Beth;&lt;/i&gt;/>
      </Submodule>
</Generic>
```

NOTE   To test that the functionality works in the portal, reference the following:

- "Registering a Module with SIP" on page 188

- "Restarting the Servlet Engine" on page 191

- "Adding the Module to a Portal View" on page 191

- "Testing a New Module" on page 192

Displaying the Contents of a URL

Through SIP, you can display the contents of a specified URL. For example, if you have an HTML file on a different server, or a CGI program, you can display the URL by referencing it in a portal view file.

You can display URLs in one of three ways:

- Inline frames (IFRAMEs). Displays in scrollable window. The browser bears the responsibility for displaying the URL. Displays a link on browsers that do not support IFRAMEs.

- Anchor tags. Displays a link.

- Embedded content. Displays in a non-scrollable window the content of the URL. SIP takes control of the content for display purposes.

To display the contents of a URL, define the Url element in a Generic element. See the examples in Figure 7-10 on page 175 and the element definitions in Table 7-9 on page 206.

All three mechanisms can be used by the URL submodule to pass a parameter (user name, role properties, SIP home directory) to the application. You can also make it part of the path to the file or URL that you are accessing.

Figure 7-10   **Example of Displaying Content of a URL**

```
<Generic>
 <Submodule>
  <TitleBar title="HP E-Services IFRAME on IE,
     Link on Netscape"/>
  <Url anchorText="HP E-Services" displayMethod="inline"
     href="http://www.hp.com/e-services" inlineHeight="300"
     proxy="no"/>
 </Submodule>
</Generic>
```

Figure 7-11   **NNM Commands Integration**

```
<Generic>
  <Submodule>
    <TitleBar title="SNMP Query"/>
    <Url href="http://$NNM/OvCgi/webappmon.exe?$report&amp;sel=$target"
        displayMethod="inline" inlineHeight="300">
      <TextParm name="NNM" prompt="NNM Management Server:" value="islandia"/>
      <TextParm name="target" prompt="Target:" value="islandia"/>
```

```
     <OptionParm name="report" prompt="Report:"
                 value="Show Capabilities">
        <Option name="Demand Status Poll"
                value="app=IP+Demand+Poll&amp;act=demandStatusPoll"/>
        <Option name="Demand Poll"
                value="app=IP+Demand+Poll&amp;act=demandPoll"/>
        <Option name="Show Capabilities"
                value="app=IP+Demand+Poll&amp;act=capsPoll"/>
        <Option name="Ping" value="app=IP+Tables&amp;act=ping"/>
        <Option name="Remote Ping"
                value="app=IP+Tables&amp;act=rping"/>
     </OptionParm>
   </Url>
  </Submodule>
</Generic>
```

If you want to activate the use of inline frames through the IFRAME HTML tag, set the displayMethod attribute to inline and use the inlineHeight attribute to specify the height of the inline frame. The default is 100 pixels.

NOTE   Netscape versions prior to 6.0 do not support inline frames; the URL will be displayed as an anchor tag using anchorText as the text for the anchor. If you set displayMethod to anchor, an anchor tag will always be present on both Internet Explorer and Netscape.

NOTE   To test that the functionality works in the portal, reference the following:

- "Registering a Module with SIP" on page 188

- "Restarting the Servlet Engine" on page 191

- "Adding the Module to a Portal View" on page 191

- "Testing a New Module" on page 192

Transforming XML Into a Displayable Format Using XSLT

Through SIP you can display content from XML that is transformed using XSL (either a URL or a file). For example, if you have data in XML format that is generated from a reporter engine or a CGI program and that you typically present using a XSL transformation, SIP can transform this data for you and display it in the portal.

Figure 7-12   Examples of Displaying Transformed XML Content

```
<Generic>
  <Submodule>
    <XmlTranslation
       xmlHref="$SIP_HOME_DIRhtdocs/C/demo/nethealth.xml"
       xslHref="$SIP_HOME_DIRconf/styles/netHealthSum_pda.xsl"
       type="text/html"/>
  </Submodule>
</Generic>
<Generic>
  <Submodule>
    <TitleBar title="XML data with XSLT"/>
    <XmlTranslation xmlHref="C:/temp/todayData.xml"
         xslHref="C:/temp/topPerformers.xsl" type="text/html">
    </XmlTranslation>
  </Submodule>
</Generic>
```

NOTE   To test that the functionality works in the portal, reference the following:

- "Registering a Module with SIP" on page 188
- "Restarting the Servlet Engine" on page 191
- "Adding the Module to a Portal View" on page 191
- "Testing a New Module" on page 192

Proxying Capabilities

The SIP proxy capabilities will allow you to (1) Provide restricted access to a web page, web data, web server, and so forth, and (2) Provide access through SIP to management product data without requiring the user to log in to the product.

TIP

Read the whitepaper *Proxy Configuration* to understand how you can configure the proxy to work for your application. This white paper is available from the OpenView website.

The Generic module provides limited proxying capabilities when displaying the content of a URL. When the proxy attribute is set to yes, the SIP server will access the URL directly (over the secure network, for example). This way you can display portal data that otherwise would not be accessible to the end user. In addition, you can also provide the name and password that will be used when accessing the protected data (see auth attribute in Table 7-9 on page 206).

Not all URLs will proxy well through the Generic module. If the URL uses JavaScript extensively, the page may not display fully correctly or some of its functionality will not be available in the module. In this case, you may need to set the proxy attribute to no and make the data system accessible to end users.

Generic Module Single Sign-on

Some URLs can only be accessed after authentication has occurred. When configuring SIP to display the contents of a URL, you can also configure SIP to automatically perform the required authentication prior to accessing the target URL. This is often referred to as "Single Sign-on." SIP supports two forms of authentication for Generic module URL access: web server authentication and login page authentication.

Web Server Authentication

SIP performs web server authentication prior to accessing the URL specified in the `href` attribute if the following is true:

- Proxying is enabled (the `proxy` attribute is "yes" or `displayMethod` equals "embedded"), and

- The `auth` attribute is specified. For Web Server Authentication, the `auth` string should be of the form `"name:password"`.

Login Page Authentication

With login page authentication, SIP performs authentication by sending data to a URL that processes information normally submitted from a login page. Configuration of this feature requires some understanding of HTML and possibly javascript. In some cases, it may require an understanding of an application's internals. Login page authentication is performed if the following is true:

- Proxying is enabled (the `proxy` attribute is "yes" or `displayMethod` is "embedded"), and

- The `loginUrl` attribute is set, specifying a URL for login page processing. This URL can often be determined by viewing the HTML source of the application's login page. SIP will perform an `HTTP POST` request to this URL, sending the parameters specified in the `auth` attribute (see below).

- The `auth` attribute is set, specifying the parameters to be passed to the `loginUrl` target. These parameters can often be determined by viewing the HTML source of the application's login page. Individual parameters are separated by `&` characters.

- If `handShake` is set to "yes" and a `handShake` URL is specified, then this URL is accessed initially to get any cookies set by the website. This cookie is then used to further access the website.

See the `OVDefaultServiceDeskSpec.xml` in the `registration/defaults` directory for an example of a generic module that uses login page authentication.

Also see the `OVDefaultPerformanceInsight.xml` for an example that uses handshake.

Giving Your Module Access to SIP Data Through Variable Substitution

Your Generic-based module can be extended to allow the retrieving of user-specific information from SIP using predefined variables. If placed in the `commandLine`, `fileName`, `auth`, `href`, `xmlHref`, and `xslHref` attributes, the variables will be replaced by the Generic-based module with current SIP data.

Three variables give your integrated module access to information about the SIP home directory, the user that is logged in, and the role of that user. Note that you can access this information, but you cannot change it.

Table 7-2   **Variable Substitution**

Variable	Description
`$SIP_HOME_DIR`	This variable will be replaced with the SIP installation directory, for example, `C:\Program Files\HP OpenView\SIP`. This variable can be placed in a `commandLine`, `fileName`, `href`, `xmlHref`, `xslHref`, or `auth` attribute in the `Command`, `File`, `Url`, and `XmlTranslation` elements respectively. For an example of using `$SIP_HOME_DIR`, see Figure 7-17 on page 183.
`$OVLOGIN`	This variable allows you to make a user login name part of a command, file, XML translation, auth string, or URL. `$OVLOGIN` will be replaced with the user's login name. For examples of using `$OVLOGIN`, see Figure 7-13, Figure 7-14, and Figure 7-15 on page 182.

`$OVROLE`	Allows applications to get data about user's current role, edit permissions, organizations defined by the management data filter, and the property name. An application can use this information to determine what to display to the user.
	
	This is done by first defining a `Role` in the user role package file. Specific keywords allow you to access four aspects of the role definition: current role, edit permissions, `MgmtData` filter, and properties:
	
	`$OVROLE[OVName]`
	`$OVROLE[OVEditLevel]`
	(See Figure 7-18 and Figure 7-19 on page 183.)
	
	`$OVROLE[OVOrgs]`
	(See Figure 7-20 on page 184.)
	
	`$OVROLE[<property name>]`
	See Figure 7-21 and Figure 7-23 on page 185.

Login Name Substitution

You can substitute in a user login name as part of a command, file name, URL, XML translation, or auth string. If you have an accessible URL that is login-name based, you can substitute in $OVLOGIN as part of the URL.

If you have a CGI program that you want to refer to on another server and the program is keyed off of a user login name, you can substitute $OVLOGIN as part of the URL string. (See example in Figure 7-13.) When the Generic-based module sees this href with $OVLOGIN, it substitutes in the user name as part of the href and then allows the browser to fetch the document. In addition, you can use $OVLOGIN in an auth attribute.

Figure 7-13   Example of Login Name Substitution ($OVLOGIN in URL)

```
<Url href="http://webpage.mycompany.com/$OVLOGIN/employeephoto.gif"/>
```

In the example in Figure 7-13, the Generic-based module would display an image from the directory named for the SIP user. For example, if the SIP user name is "jan" then the accessed URL would be `"http://webpage.mycompany.com/jan/employeephoto.gif"`

Figure 7-14   Example of Login Name Substitution ($OVLOGIN in Command)

```
<Command commandline="ping $OVLOGIN"/>
```

In the example in Figure 7-14, the Generic-based module would attempt to ping a host with the same name as the logged-in user. For example, if the SIP user name is "john" then the command executed would be "ping john".

Figure 7-15   Example of Login Name Substitution ($OVLOGIN in File Name)

```
<File fileName="C:/EmployeeFiles/$OVLOGIN.html"/>
```

In the example inFigure 7-15, the Generic-based module would display a file named for the SIP user. For example, if the SIP user name is "Jan" then the accessed file would be "C:/EmployeeFiles/Jan.html".

Figure 7-16   Example of Login Name Substitution ($OVLOGIN in auth attribute)

```
<Url href="http://webpage.mycompany.com/"
    auth="$OVLOGIN:passwd"/>
```

Figure 7-16 is an example of web server authentication accomplished with variable substitution.

SIP Installation Directory Substitution

In the example in Figure 7-17 on page 183, the portal will display a file called services.html that is located in the SIP home directory hierarchy.

For example, if SIP is installed in:
C:\Program Files\HP OpenView\SIP

Then the displayed file is:
C:\Program Files\HP OpenView\SIP\htdocs\C\demo\services.html

Figure 7-17   **Example of $SIP_HOME_DIR Used in File Name**

```
<Submodule>
    <TitleBar title="Services Purchased"/>
    <File fileName="$SIP_HOME_DIRhtdocs/C/demo/services.html"
        type="text/html" stripHtmlHeader="yes"/>
```

User Role Substitution

In the example in Figure 7-18, the portal will execute myapplication and pass the user's current role and edit permissions as two parameters to myapplication. The application can then use this information to determine what to display to this user.

Figure 7-18   **Example of $OVROLE[OVNAME] Used in a Command**

```
<Command commandline="myapplication $OVROLE[OVName] $OVROLE[OVeditLevel]"/>
```

If the role is AcmeTechnical and the edit permissions level is ViewAdmin, then the command executed would be:

myapplication AcmeTechnical ViewAdmin.

Figure 7-19   **Example of $OVROLE[OVNAME] Used in a URL href**

```
href="http://webpage.mycompany.com/foo.exe?role=$OVROLE[OVNAME]&amp;permission=$
OVROLE[OVeditLevel]"
```

In the example in Figure 7-19, the portal will execute the remote application and pass the user's current role and edit permissions as two parameters to foo.exe. If the role is AcmeTechnical and the edit permissions level is ViewAdmin, then the URL accessed would be:

```
"http://webpage.mycompany.com/foo.exe?role=AcmeTechnical&amp;permission=ViewAdmin"
```

Figure 7-20   Example of $OVROLE[OVOrgs] Used in a Command

```
<Command commandline="myapplication '$OVROLE[OVOrgs]'"/>
```

In the example in Figure 7-20, the portal will execute myapplication and pass the management data associated with the user's current role as a parameter to myapplication.

If the management data filter defines the three organizations Ariston, Kalloi, and Another Group, then the command executed would be:

```
myapplication 'Ariston|Kalloi|Another Group'
```

Rules for Substitution of Management Data

- If a role is allowed to see all management data, the value of $OVROLE[OVOrgs] will be replaced with "<ALL>"

- If a role is not allowed to see any management data, the value of $OVROLE[OVOrgs] will be replaced with "<NONE>"

- If a role is allowed to see one organization's data, the value of $OVROLE[OVOrgs] will be replaced with the organization name.

- If a role is allowed to see the data of multiple organizations, then the organization names are passed as a string separated by the pipe character. For example, "Ariston|Kalloi|Another Group".

Figure 7-21   Example of Role Properties Defined in User Role Package File

```
<Properties>
  <Property name="MgmtApp.login" value="acme/>
  <Property name="MgmtApp.password" value="acmepasswd"/>
</Properties>
```

In the user role package file, you would define the role properties as in Figure 7-21. Then, in the Generic-based module code, you would define the URL using $OVROLE[<property name>]. In the example in Figure 7-23, the portal will execute the remote application and pass the user's MgmtApp login and password as two parameters to MgmtApp.exe.

NOTE	A module, whether supplied or Generic-based, may have need for additional role attributes. The role properties element can contain any number of property elements, and each property element has two attributes: name and value. The value of the name attribute is property name. The value of the value attribute is property value.

For example, the supplied OVIS module uses Role Properties to map a SIP role to a OVIS server and customer. To do this, you must define two properties named: OVIS.server and OVIS.customer. The value of the OVIS.server should be set to the fully qualified name of the OVIS system. The value of OVIS.customer should be set to the OVIS customer name. For detailed instructions on configuring role properties for OVIS and OVSD, see the individual SIP manuals that handle the integrations of those products.

Generic-based modules can also access the additional role properties. If your module needs to access role properties information, define the role properties in the role file and use variable substitution to pass the information to your application.

Figure 7-22   Example of Login Name Substitution ($OVLOGIN in auth attribute)

```
href="http://webpage.mycompany.com/" auth="$OVLOGIN:acmepasswd"
```

Figure 7-16 is an example of web server authentication accomplished with variable substitution.The resulting auth string would look like this:

```
acme:acmepassword
```

Figure 7-23   Example of $OVROLE[<property name>] Used in a URL

```
<Url
href="http://webpage.mycompany.com/MgmtApp.exe?login=$OVROLE[MgmtApp.login]&amp;password=$OVROLE[MgmtApp.password]"/>
```

Referring to the Properties definition in the user role package file in Figure 7-21, the accessed URL would be:

```
http://webpage.mycompany.com/MgmtApp.exe?login=acme&amp;password=acmepasswd
```

Figure 7-24　　　　**Example of $OVROLE in Login Page Authentication**

```
auth="vLogin=$OVROLE[ServiceDesk.userName]&amp;vPasswd=$OVROLE[ServiceDesk.password]&amp;v
HaveAccount=yes&amp;vTimezone=$Timezone&amp;vLanguage=$Language"
```

Figure 7-24 is an example of login page authentication accomplished using variable substitution. For a example definitions of the other elements required to make login page authentication work, see the default Service Desk module:

Windows:
`%SIP_HOME%\registration\defaults\OVDefaultServiceDeskf4.xml`

UNIX: /opt/OV/SIP/registration/defaults/OVDefaultServiceDeskf4.xml

Adding an Edit GUI to Your Module

An edit GUI can be added to individual submodules or to the module as a whole. An edit GUI can be valuable for `Command`, `File`, `Url`, and `XmlTranslation` submodules. With this feature, you can define parameters and then reference them in the `commandLine`, `fileName`, `href`, `xmlHref`, or `xslHref`. In the reference to the parameter, the tag must be $<*ParameterName*> where <*ParameterName*> is the name of the parameter defined in the element.

For examples, see Figure 7-4 on page 168, Figure 7-7 on page 171, and Figure 7-5 on page 168. Also, see examples in the following directory:

Windows: `%SIP_HOME%\integrations\`
UNIX: /opt/OV/SIP/integrations/

Refer, also, to the `TextParm` attributes and `OptionParm` attributes in the section "Generic Module DTD" on page 193.

Adding Status Text, Graphics, and Links to a Submodule Title

There are four types of information that can be added to an submodule title bar area. At a minimum, you typically have a title using the `title` attribute. In addition, you can have a link to detailed data, an area for status text, and an area for a status image.

The submodule title is filled in from left to right in the following order:

- `title`
- `titleAnchorUrl`|`titleAnchorText`
- `status`
- `statusImageUrl`/`statusImageAltText`

Figure 7-25   **Titlebar Attributes of a Submodule**
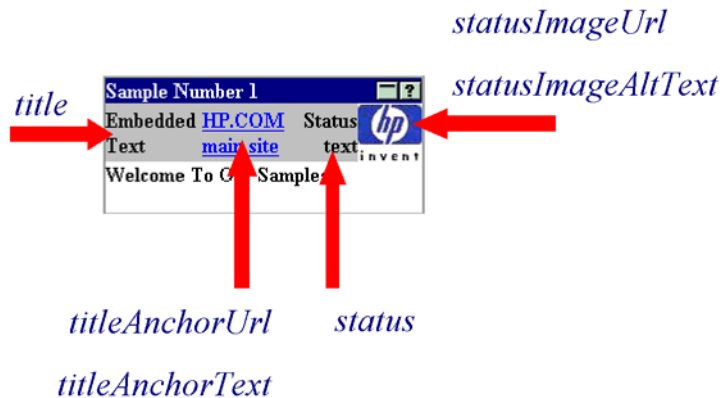
NOTE	For detailed information on elements and attributes of the Generic module, see "Generic Module DTD" on page 193, Table 7-3 on page 196, and Table 7-6 on page 198.

Adding a Link in the Submodule Title Bar

There are two attributes of the `TitleBar` element that control placing a link to detailed data in the title bar. These are `titleAnchorUrl` and `titleAnchorText`. Specify the link's URL in `titleAnchorUrl` and specify the link's text in `titleAnchorText`.

For example:

```
<TitleBar titleAnchorText="Server Health"
   titleAnchorUrl="/OvSipDocs/C/demo/server_details.html"/>
```

Adding Status Text and Status Graphics in the Submodule Title Bar

Plain status text can be presented in the submodule title bar using the `status` attribute. In addition, an image can be placed in the submodule title using the attributes: `statusImageUrl` and `statusImageAltText`. The following example demonstrates both:

```
<TitleBar status="115 alarms"
          statusImageAltText="80% major, 20% Normal"
          statusImage="/OvSipDocs/C/demo/alarm1.gif"
         title="Threshold Alarms"/>
```

Adding Online Help to a Generic-Based Module

SIP provides the capability to add online help for a Generic-based module. When the user clicks the "?" button on the module title bar, the Generic module will display the specified HTML help file. You can provide a help topic for the default module and also override it for an individual portal view. For instructions, see "Adding and Customizing Module Help Topics" on page 157.

Registering a Module with SIP

After you have created your module you can register it with SIP. This is done through a module registration file. The registration file defines such information as a unique ID for the module, a reference to the servlet, an associated help file for the module, and module capabilities.

1. Create a module registration file by copying, renaming, and modifying one of the existing ones located in the following directory:

   Windows: `%SIP_HOME%\registration`
   UNIX: `/opt/OV/SIP/registration`

2. Edit the registration file to define module-specific information:

   - Enter `title`. This is the title that is displayed in the module's title bar in the portal.

   - Enter `classid`. This required attribute uniquely identifies the module. It must be unique among all registered modules.

   - Enter `implementation`. This required attribute is a reference to the servlet that implements the module. This is the name of the Java class implementing the Generic servlet. For Generic modules, this value is always the same:

   `"/servlet/com.hp.ov.portal.modules.ovgeneric.GmGenericServlet"`

   For an example, see Figure 7-26 on page 190.

   - Enter `outputType`. This attribute is used to determine in what column type to place the module.

   - Set the `add` attribute to "yes" if you want to allow the adding of this module from the user interface. If "yes", SIP will check the user configuration file to determine whether to allow it.

   - Set the `edit` attribute to "yes" if you want to allow editing capabilities from the user interface. If "yes", SIP will check the user configuration file to determine whether to allow it.

   - Enter `help`, a pointer to the default help file for this module.

   - Enter `configDTD`, the name of the module's Document Type Definition (DTD). This is the `OVGeneric.dtd`.

   - Enter a value for `defaultConfigXML` only if `add="true"`. This optional attribute provides the path to an XML file with an XML fragment for the module configuration. This XML fragment will be added to the portal view when a user with editing permissions set to `ViewAdmin` adds the module through the module selection `Add` button.

3. Save the registration file.

Figure 7-26   Sample Module Registration File from Integrations Directory

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

<ModuleRegistration
   vendorName="Hewlett-Packard Company"
   vendorURL="http://www.openview.hp.com"
   description="HP OpenView SIP Keynote Module"
   version="1.0"
   title="Keynote"
   classid="Keynote"
   implementation="/servlet/com.hp.ov.portal.modules.ovgeneric.GmGenericServlet"
   outputType="wide"
   add="yes"
   edit="no"
   help="/OvSipDocs/C/help/<mod_directory>/<topic.html>"
   defaultConfigXML="defaults/OVDefaultKeynote.xml"
   configDTD=""
/>
```

Figure 7-27   OVModuleRegistration.dtd

```
<!-- OVModuleRegistration.dtd -->
<!-- Copyright (c) 2000 Hewlett-Packard Company -->
<!-- $Revision: /main/BACCHUS/8 $ -->
<!-- $Date: 2001/03/15 00:13 UTC $ -->

<!ELEMENT ModuleRegistration EMPTY>

<!ATTLIST ModuleRegistration
   vendorName          CDATA #IMPLIED
   vendorURL           CDATA #IMPLIED
   description         CDATA #IMPLIED
   version             CDATA #IMPLIED
   category            CDATA "General"
   title               CDATA #IMPLIED
   classid             ID    #REQUIRED
   implementation      CDATA #REQUIRED
   outputType       (narrow|NARROW|wide|WIDE|any|ANY) "narrow"
   add                 (1|0|YES|yes|NO|no) "no"
   edit                (1|0|YES|yes|NO|no) "no"
   help                CDATA #IMPLIED
   configDTD           CDATA #IMPLIED
   defaultConfigXML    CDATA #IMPLIED
   moduleTimeout       CDATA "90">
```

Restarting the Servlet Engine

After registering a module with SIP, you must stop and restart the servlet engine before the changes take effect in SIP. You can do so through the `SIP Administration Pages` or from the command line. For instructions, see "Restarting the Servlet Engine" on page 291.

Adding the Module to a Portal View

After you create and register your Generic-based module, you can add it to a portal view so that it appears in a portal. Generic-based modules are added in the same way other modules are added: (1) Through the interface (the preferred method), or (2) Through direct editing of the portal view XML file to insert the `Generic` and `Submodule` elements and attributes.

Through the SIP Interface (Preferred Method)

1. Log in to SIP as a user with access to the appropriate role. If the user has access to multiple roles, switch to the appropriate one.

2. Create or go to the tab to which you want to add the module.

3. Scroll to the bottom of the tab.

4. Select a module from the drop-down list and click `[Add]`.

Through Direct Editing of the XML

Generic-based modules can be added and configured by directly editing a portal view file. This approach is not recommended because you can easily introduce XML syntax problems.

Make a backup of configuration 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.

NOTE   For detailed information on elements and attributes of the Generic module, see "Generic Module DTD" on page 193, Table 7-3 on page 196 and Table 7-6 on page 198.

1. In an ASCII editor, open the file that has your Generic-based module code, and copy the entire `Generic` element. The XML files for default modules are stored in the following location:

   Windows: `%SIP_HOME%\registration\defaults\`
   UNIX: `/opt/OV/SIP/registration/defaults/`

2. Open the portal view file to which you want to add the Generic-based module. Portal view files are located in following directory:

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

3. Find the `Sheet` (tab) and `Column` that you want the Generic module to appear on, and create a new `ModuleInstance`. For information on the `ModuleInstance` element, see "Portal View DTD" on page 142.

NOTE   Make sure the new module instance has an `id` that is unique among all module instances in the portal view file.

4. Paste the `Generic` element into the `ModuleInstance` element.

5. After modifying XML files, always validate the XML syntax. For help on validation tools, see "Validating XML Files" on page 313.

Testing a New Module

It is important to test the new module to make sure the functionality is working as expected. If you added the module manually to the portal view file, verify that the syntax is correct and that the executables and files are in locations that are accessible to the SIP.

1. Log in to SIP as a user with access to the appropriate role. If the user has access to multiple roles, switch to the one that displays the module you added.

2. Verify that the Generic-based module displays and functions as intended. Make sure the results in the browser are correct.

3. Modify the Generic module XML, as needed.

Generic Module DTD

To create integrated modules based on the Generic module, you need to understand the document type definition (DTD) and specifications behind the module. Each instance of the Generic module is configured based on the DTD defined in the OVGeneric.dtd file located in:

Windows: %SIP_HOME%\conf\share\views\
UNIX: /opt/OV/share/SIP/conf/share/views/

A Generic element defines an instance of the Generic module and contains the following:

```
<Generic> consists of one or more:
  <Submodule> consists of:
    <TitleBar>
    <Command> or <File> or <URL>
    <EmbeddedHtml>
  </Submodule>
</Generic>
```

Figure 7-28 on page 193 shows the Generic module DTD. Table 7-3 through Table 7-14 describe the DTD elements and attributes.

Figure 7-28   OVGeneric.dtd

```
<!-- OVGeneric.dtd -->
<!-- Copyright (c) 2000 Hewlett-Packard Company -->
<!-- $Revision: /main/BACCHUS/HAITI/ARUBA/3 $ -->
<!-- $Date: 2002/12/03 18:24 UTC $ -->

<!ELEMENT Generic (GlobalParms?, Submodule+)>
<!ELEMENT GlobalParms ((TextParm*, OptionParm*)*)>
<!ELEMENT Submodule (TitleBar?, (Command? | File? | Url?
                 | XmlTranslation?), EmbeddedHtml?)>
<!ELEMENT TitleBar EMPTY>
<!ATTLIST TitleBar
      title CDATA #IMPLIED
      titleAnchorUrl CDATA #IMPLIED
      titleAnchorText CDATA #IMPLIED
      status CDATA #IMPLIED
      statusImageUrl CDATA #IMPLIED
      statusImageAltText CDATA #IMPLIED
>
<!-- EmbeddedHtml - this is HTML that will be embedded as -->
<!-- it is at the end of the extension i.e. below command output -->
```

```
<!-- or file. Notice also that Submodule may display only -->
<!-- Embedded HTML if an Submodule doesn't containt Command, -->
<!-- File nor Url -->

<!ELEMENT EmbeddedHtml EMPTY>
<!ATTLIST EmbeddedHtml
      data CDATA #IMPLIED
>
<!ELEMENT Command ((TextParm*, OptionParm*)*)>
<!ATTLIST Command
      commandLine CDATA #REQUIRED
      type CDATA "text/plain"
      expires CDATA "0"
      stripHtmlHeader (yes | YES | no | NO | 0 | 1) "no"
>

<!ELEMENT File ((TextParm*, OptionParm*)*)>
<!ATTLIST File
      fileName CDATA #REQUIRED
      type CDATA "text/plain"
      encoding CDATA "UTF-8"
      displayFileInfo (yes | YES | no | NO | 0 | 1) "no"
      stripHtmlHeader (yes | YES | no | NO | 0 | 1) "no"
>

<!-- If displayMethod is "inline" or "anchor" then to proxy -->
<!-- or not depends "proxy" attribute. If displayMethod is -->
<!-- "embedded" then we are always proxing (proxy should -->
<!-- be set to "yes", but actually portal ignores this attribute. -->
<!-- auth and loginUrl are used only if proxying is being done. -->
<!-- If auth is set, but loginUrl is not, web server -->
<!-- authentication is performed.  In this case, auth should -->
<!-- be of the form "user:password". If both auth and loginUrl are set, -->
<!-- authentication is performed via the loginUrl, which specifies -->
<!-- the URL for a login page.  In this case, auth should contain the -->
<!-- URL parameters necessary to perform the login, separated by -->
<!-- "&amp;". Usually the parameter string contains some sort of -->
<!-- user and password specification. If handshake is set to "yes" then a -->
<!-- handshake is done initially, i.e., the handshakeUrl is accessed. If any -->
<!-- cookie is available in the response, this cookie is used when making -->
<!-- further requests to the website. Handshake only works for web server -->
<!-- authentication. -->

<!-- The "windowName" attribute identifies the name of the window into -->
<!-- which the content of the Url should be displayed. The "showAsButton" -->
<!-- attribute applies only if displayMethod is "anchor". It causes the -->
<!-- Url to be shown as a button at the bottom of the module instead -->
<!-- of a link. -->
```

```
<!ELEMENT Url ((TextParm*, OptionParm*)*)>
<!ATTLIST Url
      href CDATA #REQUIRED
      displayMethod (inline | anchor | embedded) #REQUIRED
      inlineHeight CDATA "100"
      ifNoIFramesMsg CDATA #IMPLIED
      anchorText CDATA "click here"
      proxy (yes | YES | no | NO | 0 | 1) "no"
      auth CDATA #IMPLIED
      handshake (yes | YES | no| NO | 0 | 1 ) "no"
      handshakeUrl CDATA #IMPLIED
      windowName CDATA #IMPLIED
      showAsButton (yes | YES | no | NO | 0 | 1) "no"
>

<!-- "xmlHref" and "xslHref" can be a URL or a file -->
<!-- "type" is a MIME type of the output generated after applying -->
<!-- XSL transformation to XML data -->
<!-- for example, "type" can be set to "text/plain" or "text/html" -->

<!ELEMENT XmlTranslation ((TextParm*, OptionParm*)*)>
<!ATTLIST XmlTranslation
      xmlHref CDATA #REQUIRED
      xslHref CDATA #REQUIRED
      type CDATA #REQUIRED
>

<!ELEMENT TextParm EMPTY>
<!ATTLIST TextParm
      prompt CDATA #REQUIRED
      name CDATA #REQUIRED
      value CDATA #IMPLIED
>

<!ELEMENT OptionParm (Option+)>
<!ATTLIST OptionParm
      prompt CDATA #REQUIRED
      name CDATA #REQUIRED
      value CDATA #IMPLIED
>

<!ELEMENT Option EMPTY>
<!ATTLIST Option
      name CDATA #REQUIRED
      value CDATA #REQUIRED
>
```

Table 7-3   **Generic Element**

Element	Description
Submodule	A child element of the `Generic` element, `Submodule` contains zero or 1 `TitleBar` elements, zero or 1 of the following elements: `Command`, `File`, or `Url`; and zero or 1 `EmbeddedHtml` element.
GlobalParms	A child element of the `Generic` element, `GlobalParms` gives you a way to define parameters that apply to all submodules within the Generic module.

Table 7-4   **GlobalParms Elements**

Element	Description
TextParm	An optional child element of the `GlobalParms` element. When defined within the GlobalParms element `TextParm` element defines a parameter that will be applied to all submodules within a generic module. The Portal will provide the UI for an end user to edit this parameter (module must be editable).
OptionParm	An optional child element of the `GlobalParms` element. `OptionParm` element defines an option parameter that will be applied to all submodules within a generic module. The option parameter is a parameter with predefined values (options) that a user can choose from. The Portal will provide the UI for a user to choose the value of this parameter from the drop-down list (module must be editable). `OptionParm` element must contain at least one `Option` element.

Table 7-5   **Submodule Elements**

Element	Description
TitleBar	An optional child element of the `Submodule` element, `TitleBar` displays a gray titlebar at the beginning of a Submodule. A Submodule can have one `TitleBar`.

Table 7-5   Submodule Elements (Continued)

Element	Description
File	An optional child element of the Submodule element, File imports a file on your local machine or network. It does not access files that require HTTP access. File can be HTML (text/html) or plain text (text/plain). For more information see "Displaying HTML from an External File" on page 170.
Command	An optional child element of the Submodule element, Command displays the output of any executable command that generates stdout. For more information, see "Displaying the Output from a Command" on page 167. Note that parameter substitution can be used with the Command element, as described in "Giving Your Module Access to SIP Data Through Variable Substitution" on page 180.
Url	An optional child element of the Submodule element, this element displays the content of a URL. Using Url, you can display the contents of a remote file that requires HTTP access, or execute a CGI program. For more information, see "Displaying the Contents of a URL" on page 175.
XmlTranslation	An optional child element of the Submodule element, this element transforms XML data with XSL and displays the results. XML and XSLT files can reside on your local machine or network, or they can be accessed remotely using HTTP. For more information, see "Transforming XML Into a Displayable Format Using XSLT" on page 177.
EmbeddedHtml	An optional element that embeds raw HTML. Can be used by itself or in conjunction with command, file, or URL. If used with one of those, the value of EmbeddedHtml is output after the processing of the command, file, or URL and must be defined in the XML file after the command, file, or URL. For more information, see "Displaying Embedded HTML" on page 173.

Table 7-6   TitleBar Attributes

Attributes	Description
title	The name displayed in the submodule title bar (e.g., "Billing" or "Trouble Tickets"). The value can be either simple text or HTML. If HTML is used, characters such as "<" and ">" need to be appropriately escaped according the rules for valid HTML.
titleAnchorUrl	Defines a URL ("http:...") to be accessed through the submodule titlebar. The attribute titleAnchorText must also be set in order for this attribute to have an effect.
titleAnchorText	Defines the text to be placed within the anchor tag and displayed in the submodule title bar. The attribute titleAnchorUrl must also be set for this to have an effect.
status	Status text to be displayed on the right-hand side in the submodule title. The value can be either simple text or HTML.
statusImageUrl	Defines a URL ("http:...") of an image to display right justified in the submodule title bar.
statusImageAltText	Defines alternate text to display for the image.

NOTE   For more information on TitleBar, see "Adding Status Text, Graphics, and Links to a Submodule Title" on page 187.

Table 7-7   Command Attributes and Elements

Attributes and Elements	Description
commandLine	A required attribute, commandLine defines the command to use with a submodule.

Example:
"commandLine="/usr/sbin/ping hostname 10"

Optional arguments to a command can be placed as part of the commandLine attribute. If the value of commandLine contains one of the following, the command referenced by the attribute has the option of using this argument to determine a user context:

$SIP_HOME_DIR
$OVLOGIN
$OVROLE

For more information on substitution, see "Giving Your Module Access to SIP Data Through Variable Substitution" on page 180.

Be careful when presenting the results of a command. Unless configured differently, a command is executed as:

• *Windows:* the local "System" account

• *HP-UX:* www/www

• Solaris: nobody/nogroup

Make sure that only knowledgeable personnel implement this feature. |

Table 7-7   Command Attributes and Elements (Continued)

Attributes and Elements	Description
type	An attribute that indicates what type of output a command will generate based on standard MIME types. There are two values for this enumerated type: text/plain: The output from the command is pure ASCII text. In this mode, the Generic module will prefix the command's output with the HTML tag \<pre\> and will append the HTML tag \</pre\> after the command completes. This wrapping is necessary to support non-HTML output. text/html: The output from the command contains HTML text. No wrapping of the output is performed.
expires	An attribute that indicates the number of minutes the results of command execution are valid. The results of running a command are cached if expires is greater than 0 minutes. The cached results will be used the next time the same command is processed until the number of minutes specified in expires has elapsed. When the number of minutes has elapsed, the Generic module will reinvoke the command instead of using previous cache. The default is 0 minutes, meaning that no caching is performed and the command is always executed.

Table 7-7   **Command Attributes and Elements (Continued)**

Attributes and Elements	Description
stripHtmlHeader	An attribute that is used when the value of `type` equals "`text/html`". This attribute can be set to `yes` in order to display just the contents of the HTML `<body>` tag. All HTML header information from the command is removed from the output stream. This attribute may need to be set if the referenced command will generate an entire valid HTML document including HTML header information. It may not be valid to return more than one set of HTML header tags to the browser. It is important to remove the additional tags from the referenced command. The default value of this attribute is `no`. Possible values: `yes|YES|no|NO|0|1`.
	Output from the portal generates a complete HTML document including `<Html>`, `<Head>`, and `<Body>` tags. If the output of your command generates its own `<Html>`, `<Head>`, and `<Body>` tags, you will have more than one each of the `<Html>>`, `<Head>`, and `<Body>` tags. If this is the case, set `stripHtmlHeader` to "yes."
	The default is set to "no" to allow the functioning of any potential scripts that you have placed in the `<Head>` tag. Be aware that a document containing multiple `<Html>`, `<Head>`, and `<Body>` tags may cause unpredictable behavior in the browser. If a problem arises that you suspect is related to header data, change `StripHtmlHeader` to "yes."
TextParm	The optional child element of `File`, `Command`, `Url`, or `XmlTranslation` elements. `TextParm` element defines a parameter that can be used in `fileName`, `commandLine`, `href`, or auth attribute. The Portal will provide the UI for an end user to edit this parameter (module must be editable).

Table 7-7   **Command Attributes and Elements (Continued)**

Attributes and Elements	Description
OptionParm	The optional child element of `File`, `Command`, `Url`, or `XmlTranslation` elements. `OptionParm` element defines an option parameter that can be used in `fileName`, `commandLine`, `href`, or auth attribute. The option parameter is a parameter with predefined values (options) that a user can choose from. The Portal will provide the UI for a user to choose the value of this parameter from the drop-down list (module must be editable). `OptionParm` element must contain at least one `Option` element.

Table 7-8   **File Attributes**

Attributes	Description

fileName	A required attribute, `fileName` defines the file that can be accessed and displayed. The file needs to reside on the same system or be accessible through the network. The file path can be absolute (starting from the root directory) or you can use `$SIP_HOME_DIR` to reference files relative to the SIP installation directory.
	
	Example:
	`fileName="/tmp/file.html"`
	
	You can take advantage of a portal user context if the value of fileName contains one of the following:
	
	`$SIP_HOME_DIR`
	`$OVLOGIN`
	`$OVROLE`
	
	For more information on substitution, see "Giving Your Module Access to SIP Data Through Variable Substitution" on page 180.

Table 7-8   **File Attributes (Continued)**

Attributes	Description
type	Indicates the type of file. There are two values for this attribute: text/plain: The file content is pure ASCII text. In this mode, the Generic module will prefix the file content with the HTML tag <pre> and will append the HTML tag </pre> after the file content. This wrapping is necessary to support non-HTML files. text/html: The file contains HTML text. No wrapping of the content is performed.
displayFileInfo	This attribute can be set to "yes" in order to display the full path to the file and the time when the file was last modified in the browser. This information will precede the file content. The default value for this attribute is "no." The default is not to display the full path name or modification time.

Table 7-8   File Attributes (Continued)

Attributes	Description
stripHtmlHeader	This attribute is used to strip the `<Head></Head>` contents before displaying a file. This is useful if the imported file contains a `<Head>` section and the client browser cannot handle the reception of multiple `<Head>` elements (one from the portal and one from the file).
	When the value of `type` equals "`text/html`", this attribute can be set to "`yes`" to display just the contents of the HTML `<body>` tag. All HTML header information from the file will be removed from the output stream. This attribute may need to be set if the referenced file contains an entire valid HTML document including HTML header information. It may not be valid to return more than one set of HTML header tags to the browser. It is important to remove the additional tags from the referenced file. The default value of this attribute is "no." Possible values: `yes`|`YES`|`no`|`NO`|`0`|`1`.
	Output from the portal generates a complete HTML document including `<Html>`, `<Head>`, and `<Body>` tags. If the output of your command generates its own `<Html>`, `<Head>`, and `<Body>` tags, you will have more than one each of the `<Html>`, `<Head>`, and `<Body>` tags. If this is the case, set `stripHtmlHeader` to "yes."
	The default is set to "no" to allow the functioning of any potential scripts that you have placed in the <Head> tag. Be aware that a document containing multiple <Html>, <Head>, and <Body> tags may cause unpredictable behavior in the browser. If a problem arises that you suspect is related to header data, change `StripHtmlHeader` to "yes."
encoding	This attribute indicates the character set the file is encoded in. The default value is "`UTF-8`". Change the default value if the file uses non-ASCII characters.

Table 7-8   **File Attributes (Continued)**

Attributes	Description
TextParm	The optional child element of `File`, `Command`, `Url`, or `XmlTranslation` elements. `TextParm` element defines a parameter that can be used in `fileName`, `commandLine`, `href`, or auth attribute. The Portal will provide the UI for an end user to edit this parameter (module must be editable).
OptionParm	The optional child element of `File`, `Command`, `Url`, or `XmlTranslation` elements. `OptionParm` element defines an option parameter that can be used in `fileName`, `commandLine`, `href`, or auth attribute. The option parameter is a parameter with predefined values (options) that a user can choose from. The Portal will provide the UI for a user to choose the value of this parameter from the drop-down list (module must be editable). `OptionParm` element must contain at least one `Option` element.

Table 7-9   **Url Attributes**

Attributes	Description

`href`	A required attribute that defines the URL to use with this submodule.
	
	Examples:
	`href="http://www.hp.com"`
	
	Optional arguments to a URL can be placed as part of the `href` attribute. If the value of `href` contains one of the following, the URL referenced by the attribute has the option of using this argument to determine a user context for the URL:
	
	`$SIP_HOME_DIR`
	`$OVLOGIN`
	`$OVROLE`
	
	The default setting for port is 80, and the default protocol is http. These are configurable through the URL specified as the `href` for the module default XML file or the module instance.
	
	For more information on substitution, see "Giving Your Module Access to SIP Data Through Variable Substitution" on page 180.

`displayMethod`	A required attribute that can be set to either `inline`, `anchor` or `embedded`. The `inline` value indicates to SIP that `<IFRAME>` tags should be used. If you set the attribute to inline and the browser does not support inline frames, the browser will ignore the `<IFRAME>` tags and instead present a standard `<A>` tag. Browsers other than IE, including all Netscape browsers prior to version 6.0, do not support the `<IFRAME>` tag. The `anchor` value tells the portal to display only a link to the specified URL. The `embedded` value tells the portal to access the specified URL and embed (display) its content into the submodule.

Table 7-9   Url Attributes (Continued)

Attributes	Description
inlineHeight	This attribute applies only if displayMethod="inline". It specifies the height, in pixels, that will be provided for an inline frame. If not supplied, the default is 100 pixels.
ifNoIFramesMsg	This attribute applies only if displayMethod="inline". If the client browser does not support inline frames, this text will be the name of the link.
anchorText	This attribute indicates the text to display on browsers that do not support the <IFRAME> tag, such as versions of Netscape that are earlier than 6.0, or when displayMethod is anchor. Represents the text to display for a resulting HTML <A> anchor tag. The default value is "click here".
proxy	An attribute that indicates whether SIP should proxy the data from the specified URL. If proxy="no", then the end user's browser will fetch data from the specified location. The data should be accessible to the end user. For example, set proxy to "no" when displaying user myYahoo account.
	If "proxy" equals "yes" then the SIP server will fetch data from the specified location using HTTP request. This allows the SIP server to directly communicate with another server (data source) over a secured network. Use this method to access data that are not visible to the end user, for example Concord or NNM reports.
	If displayMethod="embedded", then proxy is assumed to be "yes."
	The default value of "proxy" attribute is "no". Possible values are: yes, no, YES, NO, 0, 1

Table 7-9 **Url Attributes (Continued)**

Attributes	Description
auth	This attribute is used if proxy equals "yes." If proxy is enabled, then the auth attribute is valid; and • If loginUrl IS NOT set, web server authentication is performed, and the "auth" attribute specifies the name and password that will be used by the SIP server when accessing the protected data. • If loginUrl IS set, login page authentication is performed. The syntax of "auth" for web server authentication requires that name and password are separated by a colon. For example, auth="name:password" For information on the syntax of the auth string for login page authentication, see "Login Page Authentication" on page 180. For an example of a module that uses login page authentication, see the Service Desk module: OVDefaultServiceDesk4.xml Default module XML files are located in the following directory: *Windows:* %SIP_HOME%\registration\defaults *UNIX:* /opt/OV/SIP/registration/defaults

Table 7-9   Url Attributes (Continued)

Attributes	Description
loginUrl	This optional attribute represents a URL to a login page. When this is set, and the auth attribute is set, the Generic module will invoke the loginUrl with an HTTP POST, sending the auth string as parameters. In this context, the "auth" string contains URL parameters rather than a simple "user:password" string as with web server authentication. The cookies returned by the web server are set when proxying the target URL specified in the href attribute. These cookies are also saved on the session and used for subsequent invocations of that URL (or its child links). If proxy is enabled, then the auth attribute is valid; and • If loginUrl IS NOT set, web server authentication is performed. • If loginUrl IS set, login page authentication is performed.
handshake	This attribute indicates whether the generic module should initiate a handshake with a website. If handshake is set to "yes" then a handshake is initiated, that is, the handshakeUrl is accessed. If any cookie is available in the response, the cookie is used when making succeeding requests to the website. Handshake only works for web server authentication.
handshakeUrl	This attribute specifies the URL for the website from which the generic module will send and receive data. The handshake attribute must be set to "yes".
windowName	This attribute specifies the name of the window into which the content should be displayed.
showAsButton	This attribute applies only if displayMethod="anchor". If showAsButton equals "yes", the link will be shown as a button on the bottom of the module rather than a link. The default value is "no."

Table 7-9   Url Attributes (Continued)

Attributes	Description
TextParm	The optional child element of `File`, `Command`, `Url`, or `XmlTranslation` elements. `TextParm` element defines a parameter that can be used in `fileName`, `commandLine`, `href`, or `auth` attribute. The Portal will provide the UI for an end user to edit this parameter (module must be editable).
OptionParm	The optional child element of `File`, `Command`, `Url`, or `XmlTranslation` elements. `OptionParm` element defines an option parameter that can be used in `fileName`, `commandLine`, `href`, or `auth` attribute. The option parameter is a parameter with predefined values (options) that a user can choose from. The Portal will provide the UI for a user to choose the value of this parameter from the drop-down list (module must be editable). `OptionParm` element must contain at least one `Option` element.

Table 7-10   EmbeddedHtml Attributes

Attributes	Description
data	The value of this attribute contains HTML for output constrained by the conventions outlined in "Rules for Embedded HTML" on page 173.

Table 7-11   XmlTranslation Attributes

Attributes	Description
xmlHref	A required attribute, xmlHref defines the XML data to display through the submodule. This value can be a URL or a file. Use URL if the XML data is on the same system or accessible through the network. The file path can be absolute (starting from the root directory). Alternatively, you can use $SIP_HOME_DIR to reference files relative to the SIP installation directory.
	Examples: xmlHref="http://machine.corp.com/myData.xml" xmlHref="C:/temp/myData.xml"
	You can take advantage of a user portal context if the value of xmlHref contains one of the following:
	$SIP_HOME_DIR $OVLOGIN $OVROLE
	For more information on substitution, see "Giving Your Module Access to SIP Data Through Variable Substitution" on page 180.

Table 7-11   XmlTranslation Attributes (Continued)

Attributes	Description
xslHref	A required attribute, xslHref defines the XSL transformation to use to display data in the submodule. The XML data specified by xmlHref is transformed by XSLT in the xslHref attribute and the resulting output is displayed in the portal.
	The value of xslHref can be a URL or a file. Use URL if the XSLT resides on a different server. Simply use the file path if the XSLT is on the same system or accessible through the network. The file path can be absolute (starting from the root directory). Alternatively, you can use $SIP_HOME_DIR to reference files relative to the SIP installation directory.
	Examples: xslHref="http://machine.corp.com/topPerformersOfTheDay.xsl" xslHref="C:/transformations/topPerformersOfTheDay.xml"
	You can take advantage of a user portal context if the value of xslHref contains one of the following: $SIP_HOME_DIR $OVLOGIN $OVROLE
	For more information on substitution, see "Giving Your Module Access to SIP Data Through Variable Substitution" on page 180.
type	A required attribute, type is a MIME type of the output generated after applying an XSL transformation to XML data. For example, type can be set to text/plain or text/html.

Table 7-12   **TextParm Attributes**

Attributes	Description

`prompt`	A required attribute. Specifies the prompt for the parameter entry. Prompt will be displayed during parameter editing and it should clearly identify the parameter for a user.
	
	For example,
	`prompt="Host name:"`
	`prompt="Enter name of NNM station:"`
	
	For more information on parameter editing, see "Adding an Edit GUI to Your Module" on page 186.

`name`	The required attribute. The name of the parameter. This name preceded by the dollar sign $ can be used in `fileName`, `commandLine`, `href`, or auth attribute. The SIP portal will replace this `$name` by the parameter value before accessing the file or the URL or executing the command.
`value`	The default value of the parameter. The parameter (`$name`) in `fileName`, `href` or `commandLine` will be replaced by this value (the value of the `value` attribute) before accessing the file or the URL or executing the command.

Table 7-13   **OptionParm Attributes**

Attributes	Description

`prompt`	The required attribute. The prompt for the parameter entry. Prompt will be displayed during parameter editing and it should clearly identify the parameter for a user.
	
	For example,
	`prompt="Host name:"`
	`prompt="Choose name of NNM station:"`
	
	For more information on parameter editing, see "Adding an Edit GUI to Your Module" on page 186.

Table 7-13   **OptionParm Attributes (Continued)**

Attributes	Description
name	The required attribute. The name of the parameter. This name preceded by the dollar sign $ can be used in `fileName`, `commandLine`, `href`, or auth attribute. The SIP portal will replace this `$name` by the parameter value before accessing the file or the URL or executing the command.
value	The name of a default option. The name of an option that define the default value of the parameter. The option with this name must be defined i.e. `Option` element with this name must be contained in this `OptionParm` element.
	The parameter (`$name`) in `fileName`, `commandLine`, `href`, or auth will be replaced by the value of this option before accessing the file or the URL or executing the command.
Option	A child element of `OptionParm` element. The `Option` element defines one of options for the parameter, that is, one of the values that the parameter can take. There must be at least one `Option` element defined (contained in) for an `OptionParm` element.

Table 7-14   **Option Attributes**

Attributes	Description
name	A required attribute of the `Option` element, specifies the name of the option. This name will be displayed in the drop-down selection list to identify the option.
value	A required attribute of the `Option` element, specifies the default value of the option. If the parameter has this option value (that is, if the option is selected), the parameter in `fileName`, `href`, or `commandLine` will be replaced by this value (the value of the `value` attribute) before accessing the file or URL or executing the command.

8 Segmenting Data by Customer Organization

Understanding Customer-to-Resource Mappings

Segmenting data by customer organization requires the use of a customer model: a mapping of customer organizations to their resources. Table 8-1 shows which management software integrates with supplied modules that:

- Use the SIP customer model

- Use their own customer model

- Do no data filtering.

Table 8-1   The Use of a Customer Model by Supplied Modules

Management Software or Module	Modules Use the SIP Customer Model	Modules Use Their Own Customer Model	Modules Do No Filtering of Management Data
Network Node Manager (NNM)	X		
OpenView Internet Services (OVIS) and OpenView Performance Manager (OVPM)		X	
OpenView Reporter (OVR)	X		
OpenView Operations (OVO), OpenView Service Navigator (OVSN), and OpenView Operations for Windows (OVO/W)	X		
OpenView Service Desk (OVSD)		X	

Table 8-1   **The Use of a Customer Model by Supplied Modules (Continued)**

Management Software or Module	Modules Use the SIP Customer Model	Modules Use Their Own Customer Model	Modules Do No Filtering of Management Data
OpenView Performance Insight (OVPI)		X	
Message Board module			X
Bookmarks module			X

Using the SIP Customer Model

Several of the OpenView integrations supplied with SIP can filter management data by customer organization, using the SIP customer model.

The SIP customer model is constructed from multiple registered sources that are either XML files or programs that generate XML. The XML must be based on the SIP Simple Customer Model (SCM), which maps *organizations* to *nodes*, *interfaces*, *non-NNM-discovered objects* (for topology maps) and *services*, and is defined in an XML DTD. Creating the customer model sources means getting customer-to-resource mappings into a format that conforms to the DTD so they can be used by SIP.

Customer model sources are defined in several XML files, or in a mix of CGIs, servlets, and XML files that you must register with SIP. SIP merges these sources, or "sub-models" into one fully-integrated customer model.

Summary of the Process of Implementing a SIP Customer Model

Three tasks (explained in the following sections) are required when setting up customer-to-resource mappings and making them available to SIP:

1. Create the SIP customer model, which is defined in several XML files or a mix of programs and files, each called a customer model source.

2. Register all sources of the customer model with SIP.

3. Verify that the correct data is being displayed.

Using an External Customer Model

For products that have their own customer model that is maintained by the management product, such as OVSD, OVPI, and OVIS, SIP takes advantage of the external customer model via special properties that can be defined for a role. To display segmented data from these products, you do not create customer model sources or a management data filter per se. Instead, you define role properties that map a SIP role to a customer organization and its resources. To learn how to create valid role properties, read the SIP manuals that document the integration of these management products.

Creating a SIP Customer Model

The SIP customer model is defined in several XML files or a mix of programs and files, each called a customer model source.

Creating Customer Model Sources

A customer model source can define any number of organizations, service lists, node lists, topology map object (non-NNM-discovered nodes) lists, and interface lists, including none at all. Furthermore, the organizations and resource lists can be defined in any order within an XML file.

You have a lot of flexibility in how you go about creating the customer model sources, but you must do two main tasks:

- Create lists of resources
- Map organizations to their resources

The task of mapping organizations to resources is described in the next section "Mapping Organizations to Their Resources" on page 220.

The task of creating lists of resources is described in the SIP manuals that cover the individual management product integrations.

What You Should Know About Customer Model Sources

- A customer model source can be an XML file that you create from scratch that does the following:

  — Maps organizations to their resources, either by listing the resources directly in the file, or by referencing lists of resources.

    For detailed information on mapping organizations to their resources, see "Mapping Organizations to Their Resources" on page 220.

  — Lists the resources without any association to organizations.

    For detailed information on creating lists of nodes, interfaces, topology map objects (non-NNM-discovered nodes), and services that can be associated with organizations, see the following:

    — *HP OpenView Operations and Service Navigator Integration with SIP* (OVO_and_OVSN_Integration.pdf).

— *NNM Integration with SIP* (`NNM_Integration.pdf`)

- A customer model source can be a supplied program, such as `getcvdata.exe` and the `NNMSimpleCustomerModel` servlet.

  Use `getcvdata.exe` if you have Customer Views for NNM and you want to leverage the Customer Views organizations-to-resource mappings in the SIP customer model.

  Use the `NNMSimpleCustomerModel` servlet to retrieve filtered information from the NNM object database (`ovwdb`) and automatically generate lists of nodes and interfaces.

  For detailed information, see *Network Node Manager Integration with SIP* (`NNM_Integration.pdf`).

- A customer model source can be a program that you create yourself that provides a mapping from an arbitrary data store or provisioning system to the required "simple customer model" format.

  Use this approach if you want to write a program that converts data from an existing data source into the required XML format for the SIP customer model. For information on developing your own customer model data source, see "Developing a Custom Customer Model Source" on page 223.

Mapping Organizations to Their Resources

In order to display management data in SIP that is segmented by customer, at least one of the customer model source files must contain associations between organizations and their resources.

This procedure assumes that you have already created your resource lists, as described in *HP OpenView Operations and Service Navigator Integration with SIP* (`OVO_and_OVSN_Integration.pdf`) and *NNM Integration with SIP* (`NNM_Integration.pdf`).

1. Create an XML file based on the `SimpleCustomerModel.dtd`. The easiest way is to copy and rename an existing one, such as `CustomerModel.xml` that is supplied with SIP:

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

   Within the XML file, the path to the `SimpleCustomerModel.dtd` in the DOCTYPE statement must correctly reference the location of DTD. SIP assumes that your customer model source XML files are in

the `/conf/share/organizations` directory. You can reference the Simple Customer Model DTD locally or remotely using the following URL examples, respectively:

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

<!DOCTYPE SimpleCustomerModel SYSTEM
"http://<SIP_Server>/ovportal/servlet/DTDServer/conf/share/organizations/SimpleC
ustomerModel.dtd">
```

2. Using an ASCII or XML editor, open the file and create mappings that associate an organization with its resources. For examples, see Figure 8-1 on page 221 and Figure 8-2 on page 222.

3. Save the XML file into the `organizations` directory.

   SIP interprets relative paths relative to the `organizations` directory.

4. Register this file as a source of customer model data. For detailed instructions, see "Registering SIP Customer Model Sources" on page 235.

Figure 8-1   **Sample Customer Model Source That Maps Organizations to Resources by Listing Resources Directly in the File**

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

<SimpleCustomerModel>

   <Organization type="customer" name="Cust1">
      <NodeList>
         <Node name="cisco4k2.cnd.hp.com"/>
         <Node name="15.2.3.23"/>
         <Node type="ov-iphost" name="cisco2522"/>
      </NodeList>
      <InterfaceList>
         <Interface name="15.0.2.33"/>
         <Interface type="ov-ipv4" name="15.2.3.23"/>
         <Interface type="ov-vpn4" name="10.2.1.1/3434"/>
      </InterfaceList>
      <TopomapObjectList>
         <TopomapObject type="Selection Name" name="Key Resources"/>
      </TopomapObjectList>

   </Organization>
   <Organization type="customer" name="Cust2">
      <ServiceList>
```

```
          <Service type="webHosting" name="Cust2WebHosting"/>
          <Service type="mail" name="Cust2Mail"/>
          <Service type="news" name="Cust2News">
              <Depth>2</Depth></Service>
      </ServiceList>
      <NodeListRef href="WebServers"/>
   </Organization>
   <Organization type="customer" name="VIC1">
      <ServiceLevel>Gold</ServiceLevel>
      <ExternalKey>vic-1001</ExternalKey>
   </Organization>
   <Organization type="provider" name="PeerISP1"/>
      <NodeList name="WebServers">
        <Node name="www.cnd.hp.com"/>
        <Node name="web.fc.hp.com"/>
       </NodeList>
</SimpleCustomerModel>
```

Figure 8-2   Sample Customer Model Source That Maps Organizations to Resources by Referencing Lists of Resources

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

<SimpleCustomerModel>

   <Organization type="customer" name="Cust1">
      <NodeListRef href="List1"/>
      <InterfaceListRef href="List2"/>
      <TopomapObjectListRef href="List3"/>
   </Organization>

   <Organization type="customer" name="Cust2">
      <ServiceListRef href="List4"/>
   </Organization>

 </SimpleCustomerModel>
```

Figure 8-3   Sample Customer Model Source That Contains Lists of Resources Only

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

<SimpleCustomerModel>

      <NodeList name="List1">
        <Node name="cisco4k2.cnd.hp.com"/>
        <Node name="15.2.3.23"/>
        <Node type="ov-iphost" name="cisco2522"/>
```

```
        <Node type="ISPNews.cnd.hp.com" name="ov-iphost"/>
        <Node type="ISPGlobalNet.isp1.com" name="ov-iphost"/>
    </NodeList>
    <InterfaceList name="List2">
        <Interface name="15.0.2.33"/>
        <Interface type="ov-ipv4" name="15.2.3.23"/>
        <Interface type="ov-vpn4" name="10.2.1.1/3434"/>
        <Interface type="15.40.10.2" name="ov-ipv4"/>
    </InterfaceList>
    <TopomapObjectList name="List3">
        <TopomapObject type="Selection Name" name="Western Region"/>
        <TopomapObject type="Selection Name" name="Eastern Region"/>
    </TopomapObjectList>
    <ServiceList name="List4">
        <Service type="webHosting" name="Cust2WebHosting"/>
        <Service type="mail" name="Cust2Mail"/>
        <Service type="news" name="Cust2News">
            <Depth>2</Depth></Service>
    </ServiceList>
</SimpleCustomerModel>
```

Developing a Custom Customer Model Source

You can develop a program that provides a mapping from an arbitrary data store or provisioning system to the required SIP customer model format. SIP is capable of receiving content for the customer model from any source that can produce XML content that adheres to the SIP customer model DTD.

SIP can be configured to read data from a local file, or any arbitrary URL that is accessible from the SIP server. The URL can be a servlet, CGI, or other technology that returns XML content via HTTP.

For example, assume that you maintain a database of customers and their related managed resources. A perl CGI script named `getMyCustomerData.pl` could be written and installed on a web server, `customerdb.generic.net`. Suppose the URL to that script is "`http://customerdb.generic.net/cgi-bin/getMyCustomerData.pl`", you could add the URL as a Customer Model Source in the `OVPortalConfig.xml` file as follows:

```
 <CustomerModelSources>
    <CustomerModel
       href="http://customerdb.generic.net/cgi-bin/getMyCustomerData.pl"/>
 </CustomerModelSources>
```

The perl cgi script could contain some code similar to this:

```
#!/usr/local/bin/perl


print "content-type: text/xml\n\n";
print "<?xml version=\"1.0\" encoding=\"UTF-8\" ?>\n";
print "<!DOCTYPE SimpleCustomerModel PUBLIC \"SimpleCustomerModel\"
\"http://server.co.com/ovportal/servlet/DTDServer/conf/share/organizations/Simpl
eCustomerModel.dtd\">\n\n";

print "content-type: text/xml\n\n";
print "<SimpleCustomerModel>";
@orgs = getOrganizationsFromDb();
foreach org (@orgs) {
      print "<Organization name=\"$org\">"
      outputOrganizationData($org);
      print "</Organization>"
print "</SimpleCustomerModel>\n";
```

Your perl CGI `outputOrganizationData` script could output the customer-to-resource mappings wrapped in XML code required by SIP.

Of course this could be written in any language using a technology that is capable of returning XML via HTTP. Parameters may be specified in the URL as well. For example, you could invoke this perl script differently from multiple SIP servers as follows:

Figure 8-4   OVPortalConfig.xml on SIP server1

```
<CustomerModelSources>
  <CustomerModel
  href="http://customerdb.generic.net/cgi-bin/getMyCustomerData.pl?server=one"/>
</CustomerModelSources>
```

Figure 8-5   OVPortalConfig.xml on SIP server2

```
<CustomerModelSources>
  <CustomerModel
  href="http://customerdb.generic.net/cgi-bin/getMyCustomerData.pl?server=two"/>
</CustomerModelSources>
```

In this case, the CGI can interpret the parameters and return different content to each server. As stated previously, this content must conform to the DTD defined in `SimpleCustomerModel.dtd` file. The DTD provides a mechanism for remote reporting of errors. If the CGI encounters an error

or warning, it can report those along with any Customer Model data, and they will be added to the appropriate SIP log file. For example, the CGI could return:

```
<SimpleCustomerModel>
    <Error msg="The database is currently unavailable"/>
</SimpleCustomermodel>
```

SIP assumes that your customer model source XML files are in the /conf/share/organizations directory. You can reference the Simple Customer Model DTD locally or remotely using the following URL examples, respectively:

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

<!DOCTYPE SimpleCustomerModel SYSTEM
"http://<SIP_Server>/ovportal/servlet/DTDServer/conf/share/organizations/SimpleC
ustomerModel.dtd">
```

SIP Customer Model DTD

The SimpleCustomerModel.dtd file is located in the following directory:

Windows: %SIP_HOME%\conf\organizations\
UNIX: /opt/OV/SIP/conf/organizations/

Table 8-2   SimpleCustomerModel Child Elements

Elements	Descriptions
Organization	The name of a customer, or a logical grouping for segmenting data.
ServiceList	Can be a child element of Organization or specified outside of any Organization. If a child element, ServiceList is a listing of services that are associated with an organization. If specified outside an Organization, it is a listing of services that can be associated with multiple organizations.

Table 8-2   SimpleCustomerModel Child Elements

Elements	Descriptions
NodeList	Can be a child element of Organization or specified outside of any Organization. If a child element, NodeList is a listing of nodes that are associated with an organization. If specified outside an Organization, it is a listing of nodes that can be associated with multiple organizations.
InterfaceList	Can be a child element of Organization or specified outside of any Organization. If a child element, InterfaceList is a listing of interfaces that are associated with an organization. If specified outside an Organization, it is a listing of interfaces that can be associated with multiple organizations.
TopomapObjectList	Can be a child element of Organization or specified outside of any Organization. If a child element, TopomapObjectList is a listing of topology map objects (non-NNM-discovered nodes, such as containers and locations) that are associated with an organization. If specified outside an Organization, it is a listing of topology map objects that can be associated with multiple organizations.

Table 8-3   Organization Attributes and Child Elements

Elements and Attributes	Descriptions
type	An optional attribute that specifies the type of customer or organization. For Customer Views data, this would have values of "customer" and "provider."
name	A required attribute that specifies a unique name for identifying the organization.
DisplayString	An optional displayable string. If this element is not present, the name attribute is used for display purposes.
ServiceLevel	An optional element that specifies the level of service (e.g., platinum, gold, silver, bronze). In Customer Views, this idea is implicit in groups of customers or organizations.
ExternalKey	An optional element that represents an external database key for the organization. This is provided to migrate the organizationExternalKey attribute stored by Customer Views.

Table 8-4   **ServiceList Attributes and Child Elements**

Elements and Attributes	Descriptions
`name`	An optional attribute that specifies a unique name for identifying the `ServiceList`. The `ServiceListRef` uses the `href` attribute to reference a particular `ServiceList` by name. If multiple `ServiceList` elements have the same name value, they are viewed as the same list of services and the definitions are merged.
`Service`	A resource that can be associated with an organization. The named service does not need to contain the definition of the service but provide a way to refer a particular service that is defined elsewhere.
`ServiceListRef href`	An optional element that can be used to have multiple organizations share the same `ServiceList` definition. A `ServiceList` referenced by a `ServiceListRef` element must be identified by an `href` attribute.

Table 8-5   **Service Attributes and Child Elements**

Elements and Attributes	Descriptions
`type`	An optional attribute that specifies the type of service (for example, "webHosting" and "e-mail." The `type` value is an arbitrary string defined by the application or module dealing with the service. It is used to filter on the list of services, so that a module only needs to deal with the services it is concerned with.

Table 8-5   Service Attributes and Child Elements

Elements and Attributes	Descriptions
name	A required attribute that uniquely identifies the service.
DisplayString	An optional displayable string. If this element is not present, the name attribute is used for display purposes.
Depth	An optional element that specifies the number that represents a depth limit for displaying information about the service. This value is used by the HP OpenView Service Navigator modules. A depth value of 0 suggests that all children of the service, and the service itself, are associated with this organization.

Table 8-6   NodeList Attributes and Child Elements

Elements and Attributes	Descriptions
name	An optional attribute that specifies a unique name for identifying the NodeList. The NodeListRef uses the href attribute to reference a particular NodeList by name. If multiple NodeList elements have the same name value, they are viewed as the same list of nodes and the definitions are merged.
Node	A resource that can be associated with an organization.

Table 8-6   **NodeList Attributes and Child Elements**

Elements and Attributes	Descriptions
NodeListRef href	An optional element that can be used to have multiple organizations share the same NodeList definition. A NodeList referenced by a NodeListRef element must be identified by an href attribute.

Table 8-7   **Node Attributes and Child Elements**

Elements and Attributes	Descriptions
type	An optional attribute that specifies the type of node name. This is specified for future extensibility. For SIP 2.0, only one value is understood by the modules: "ov-iphost". An "ov-iphost" type of name is a hostname, DNS name, or IP address.
name	A required attribute that uniquely identifies the node. If the type is "ov-iphost", the name value can be an IP hostname, a DNS name, or an IP address.
DisplayString	An optional displayable string. If this element is not present, the name attribute is used for display purposes.

Table 8-8   **InterfaceList Attributes and Child Elements**

Elements and Attributes	Descriptions
name	An optional attribute that specifies a unique name for identifying the InterfaceList. The InterfaceListRef uses the href attribute to reference a particular InterfaceList by name. If multiple InterfaceList elements have the same name value, they are viewed as the same list of interfaces and the definitions are merged.
Interface	A resource that can be associated with an organization.
InterfaceListRef href	An optional element that can be used to have multiple organizations share the same InterfaceList definition. An InterfaceList referenced by an InterfaceListRef element must be identified by an href attribute.

Table 8-9   **Interface Attributes and Child Elements**

Elements and Attributes	Descriptions
`type`	Specifies the type or "address family" of the interface.
	IP-address, or `hostname/ifAlias::ifDescr` `hostname` = fully-qualified hostname or IP address `ifAlias` = from the SNMP IF-MIB (rfc2863) `ifDescr` = the first word in the SNMP MIB-II (rfc1213) `ifDescr` string
	NOTE: Either `ifAlias` or `ifDescr` can be empty, but the combination must uniquely identify the interface.
`name`	A required attribute that uniquely identifies the interface. The format of this attribute's value is dictated by the value of the "type" attribute.
`DisplayString`	An optional displayable string. If this element is not present, the `name` attribute is used for display purposes.

**Table 8-10   TopomapObjectList Attribute and Child
                     Element**

Elements and Attributes	Descriptions
`name`	An optional attribute that specifies a unique name for identifying the `TopomapObjectList`. The `TopomapObjectListRef` uses the `href` attribute to reference a particular `TopomapObjectList` by name. If multiple `TopomapObjectList` elements have the same name value, they are viewed as the same list of objects and the definitions are merged.
`TopomapObject`	A resource that can be associated with an organization. The purpose of this element is to provide a way to display and filter non-NNM-discovered objects in the SIP Topology Map module.
`TopomapObjectListRef href`	An optional element that can be used to have multiple organizations share the same `TopomapObjectList` definition. A `TopomapObjectList` referenced by a `TopomapObjectListRef` element must be identified by an `href` attribute.

Table 8-11   TopomapObject Attributes

Elements and Attributes	Descriptions
`type`	An optional attribute that specifies the type of non-NNM-discovered node name. Only one value is understood by the Topology module: "`Selection Name`".

Table 8-11 **TopomapObject Attributes (Continued)**

Elements and Attributes	Descriptions
name	A required attribute that uniquely identifies the non-NNM-discovered node. For the name attribute, enter a value from NNM's Selection Name field.

Registering SIP Customer Model Sources

You need to register with SIP the name and location of the sources of the customer model data. This is done through the `SIP Administration Pages` that are accessed through a special `SIP Administrator` role.

Registering a Customer Model Source

Customer model sources can be defined in one or more files on the system or can be programmatically generated by CGI programs and servlets. Most likely, the sources to your fully-integrated customer model are a mix of files and programs.

1. Go to the `SIP Administration Pages` by logging in as a user who can access the special `SIP Administrator` role. Switch to the `SIP Administrator` role.

2. Click the `Customer Model` tab.

3. In the `Customer Model Configuration` segment, go to `Customer Model Sources`.

4. In the `New customer model source URL` field, type a relative file name (relative to the `conf/share/organizations` directory), an absolute file name, or a URL. See examples below:

Figure 8-6　　**Example of a Relative File Name**

```
CustomerModel.xml
```

(Note that the same syntax is used regardless of whether the `conf/share/organizations` directory is local or remote.)

Figure 8-7　　**Example of an Absolute File Name**

```
c:/temp/CustomerModel.xml
```

Figure 8-8　　**Examples of URLs**

```
http://othermachine/CustomerModel.xml
http://NNMHostname/OvCgi/getcvdata.exe
http://SIPhostname/ovportal/NNMSimpleCustomerModel
```

5. Click [Add] to add the name of the customer model source to the Customer Model Sources list. The customer model is refreshed automatically, so you do not need to force a refresh by clicking [Refresh].

NOTE   After registering your customer model sources, you can create management data filters that reference one or more customer organizations in the customer model ("Defining/Modifying Management Data Filters" on page 90). After you have created the filters, you can associate each role with a filter ("Assigning a Management Data Filter to a Role" on page 95). At that point you can display segmented management data through SIP modules.

Unregistering a Customer Model Source

To verify the data from a customer model source before you unregister it, see "Viewing a Report of One Customer Model Source" on page 241.

1. Go to the SIP Administration Pages by logging in as a user who can access the special Administrator role. Switch to the Administrator role.

2. Click the Customer Model tab.

3. In the Customer Model Configuration segment, go to Customer Model Sources.

4. In the Customer Model Sources list, select a customer model source and click [Delete].

Verifying That the Correct Data is Being Displayed

The Managed Resources module provides a view of the customer model as filtered by the management data filter. It shows you exactly what data is passing the management data filter that is associated with the current role.

The Managed Resources module displays a list of organizations that are associated with the role. It also displays lists of nodes, interfaces, and services associated with each organization.

1. Log in to SIP and switch to the role for which you want to verify data.

2. Go to the drop-down list box at the bottom of a portal tab column.

3. Select the `Managed Resources` module and click `[Add]`.

4. Look at the organization and resource data to verify that the correct data is being displayed to this role.

5. When you are finished, if appropriate, delete the module from the portal view by clicking the `[X]` button on the module title bar.

Configuring the Managed Resources Module

There are several elements and attributes of the Managed Resources module that can be configured on a per-view basis through direct editing of the XML.

- `FilteredResources details` - Indicates whether the attributes of individual resources should be displayed or not. The default value is "yes."

- `ListIfAll` - For organizations and each type of resource, in the case where all are allowed, this indicates whether the list is fully enumerated or a string is displayed, for example "All Organizations" or "All Nodes" and so forth. If this element is not present, the union of resources for all the allowed organizations will be shown.

- `DisplayByOrg` - Indicates whether to group the resources that pass the filter by the organization they are associated with.

- `type` - A Perl 5 regular expression for filtering resources by their `type` field in the customer model definition.

- `OrganizationFilter`, `NodeSelection`, and `InterfaceSelection` - These elements can be used to determine which resources the various "display filters" allow. After testing the display filter here, it can be copied to the module instance where it will be used.

NOTE A second level of management data filtering (sometime called display filtering) is allowed by some of the modules. This type of filtering is optional and allows you can refine the data on a module-by-module basis. To set up filtering at this level, see the SIP manuals that document the integration of individual management products.

To Configure the Managed Resources Module

For examples of variously configured Managed Resource modules, see:

Windows: `%SIP_HOME%\samples\views\resources.xml`
UNIX: `/opt/OV/SIP/samples/views/resources.xml`

1. Make a backup of the portal view XML file before you modify it.

2. Using an ASCII editor, open the portal view file that contains the Managed Resources module that you want to modify.

3. In the XML file, find the Managed Resources module by searching for "FilteredResources".

4. Make the desired changes.

5. Save the file.

6. After you make modifications to the XML file, validate the syntax. For more information, see "Validating XML Files" on page 313.

Configuring Refresh Rates for the SIP Customer Model

From the Customer Model tab in the SIP Administration Pages, you can force a refresh of the customer model, configure the refresh rate, and configure the refresh rate of information coming from the NNM object database.

Refreshing the Customer Model

You can force a refresh of the customer model. This can be particularly useful if, for example, you are working on a customer model source and want to immediately update the customer model with your changes.

1. Go to the `SIP Administration Pages` by logging in as a user who can access the `SIP Administrator` role.

2. Click the `Customer Model` tab.

3. In the `Customer Model Configuration` segment, go to `Refresh Rates`.

4. Click the `[Refresh]` button.

Configuring the Refresh Rate for the Customer Model

The customer model is loaded the first time it is referenced by SIP and each defined refresh period after that. The refresh rate specifies how often, in minutes, you want the customer model to be refreshed.

1. Go to the `SIP Administration Pages` by logging in as a user who can access the `SIP Administrator` role.

2. Click the `Customer Model` tab.

3. In the `Customer Model Configuration` segment, go to `Refresh Rates`.

4. Type the number of minutes you want between automatic refreshes to the customer model. If the value is "0", the customer model will only be loaded when the servlet engine is restarted, or when you click `[Refresh]`.

5. If one of your customer data sources is retrieving filtered information from the NNM object database (`ovwdb`), type the number of minutes you want between automatic refreshes of the object capabilities.

6. Save the refresh rate by clicking `[Apply]` at the bottom of the portal page.

Viewing Reports of the SIP Customer Model

Through the `SIP Administration Pages` you can view a report of one customer model source or of the fully-integrated customer model.

Viewing a Report of One Customer Model Source

This report shows the contribution to the fully-integrated customer model made by a particular source.

1. Go to the `SIP Administration Pages` by logging in as a user who can access the special `SIP Administrator` role. Switch to the `SIP Administrator` role.

2. Click the `Customer Model` tab.

3. Go to the `Customer Model Sources` segment of the `Customer Model` tab, select a source, and click `[Report]`.

Viewing a Report of the Fully-Integrated Customer Model

The Customer Model Summary Report displays the fully-integrated customer model, which is composed of the data from all customer model sources. You can choose to see the customer model for all organizations or for one particular organization. If an organization has contributions from several sources, this report shows the fully-integrated organization data from all sources.

1. Go to the `SIP Administration Pages` by logging in as a user who can access the special `SIP Administrator` role. Switch to the `SIP Administrator` role.

2. Click the `Customer Model` tab.

3. Go to the `Customer Model Summary Report` segment, select one organization or all organizations.

4. Choose report options.

5. Click `[View]`.

Exporting the SIP Customer Model

You can configure a list of destinations to which the customer model is sent every time it is reloaded (refreshed). The destinations can be file names or URLs, and URLs can be CGI programs or servlets that do something appropriate with the SIP customer model, such as updating a database for use in generating customer-segmented reports.

Registering a Customer Model Export Destination

1. In the `New customer model destinations URL` field, type a relative file name (relative to the `/conf/share/organizations` directory), an absolute file name, or a URL.

   Syntax rules that apply to the entry of new customer model sources also apply to new customer model export destinations. For examples, see

2. Click [Add] to add the name of the export destination to the `Customer model export destination` list. The export destinations are refreshed automatically, so you do not need to force a refresh by clicking [Refresh].

Figure 8-9   **Example of a Relative File Name**

```
ExportModel.xml
```

(Note that the same syntax is used regardless of whether the `conf/share/organizations` directory is local or remote.)

Figure 8-10   **Example of an Absolute File Name**

```
c:/temp/ExportModel.xml
```

Figure 8-11   **Examples of URLs**

```
http://<OVR-Server>/HPOV_IOPS/cgi-bin/repimport.exe
http://<NNM_Server>/OvCgi/ovsipexport.exe
```

Removing a Customer Model Export Destination

1. Go to the `SIP Administration Pages` by logging in as a user who can access the special `SIP Administrator` role. Switch to the `SIP Administrator` role.

2. Click the `Customer Model` tab.

3. In the `Customer Model Configuration` segment, go to `Customer Model Export`.

4. In the `Customer Model export destinations` list, select an export destination and click `[Delete]`.

9 Configuring Authentication

Introduction to Supplied Authentication Providers

SIP comes with six ready-to-use authentication providers. By default, the No Password Authentication Provider is registered with SIP. If you want to use one of the other five, you will need to do the following:

• Choose one of the supplied authentication providers.

• For some authentication providers, complete the configuration tasks.

• Register the authentication provider with SIP.

If none of these meets your needs, you may need to integrate a custom authentication provider. (See "Developing a Custom Authentication Provider" on page 261.)

Choosing and Configuring an Authentication Provider

SIP comes with six ready-to-use authentication providers. By default, the No-Password Authentication Provider is registered with SIP. The following sections describe each authentication provider and how to configure it.

Null Authentication Provider

With this external authentication provider, no authentication is performed and no user name is required. Anyone who requests the portal becomes the user "anyuser."

What the user sees depends upon whether "anyuser" is defined in the User Role model, and whether there is a default user configured in the User Role model if "anyuser" is not explicitly defined.

To use this authentication provider, simply register it with SIP. To do so, see "Registering an Authentication Provider" on page 257. An example registration follows.

Example Registration of the Null Authentication Provider

```
<Authentication
    LoginPage="/ovportal"
    AuthenticationProviderClass=
        "com.hp.ov.portal.security.NullAuthenticationProvider"
    ShowLogoutButton="no"/>
```

No-Password Authentication Provider

This authentication provider is configured out of the box. It performs no authentication and only requires a user name. The user name must match the name of a user in the User Role Model (unless the defaulting mechanisms, such as a default user, are used).

This provider is useful when you want to set up and try out portal views for different users before making SIP available to end users. This is not an authentication provider that would ever be used in a production environment.

To use this authentication provider, simply register it with SIP. To do so, see "Registering an Authentication Provider" on page 257. An example registration follows.

Example Registration of the No Password Authentication Provider

```
<Authentication
  LoginPage="/ovportal/jsp/security/userName_html.jsp"
  AuthenticationProviderClass=
  "com.hp.ov.portal.security.NoPasswordAuthenticationProvider"
  ShowLogoutButton="yes"
  LogoutPage="/ovportal/jsp/security/logout_html.jsp"/>
```

Password File Authentication Provider

This authentication provider requires a user name and password. The password is stored encrypted, using UNIX crypt(3), and the supplied password is encrypted and compared to the stored password. This is reasonably secure if the network communication channel is protected (via SSL, for example) and if the password file is not accessible outside the system.

To use this authentication provider, register it with SIP, as described in "Registering an Authentication Provider" on page 257. An example registration follows.

Example Registration of the Password File Authentication Provider

```
<Authentication
    LoginPage="/ovportal/jsp/security/login_html.jsp"
    AuthenticationProviderClass=
      "com.hp.ov.portal.security.FileAuthenticationProvider"
    ShowLogoutButton="yes"
    LogoutPage="/ovportal/jsp/security/logout_html.jsp"/>
```

Configuring User Logins

For the Password File Authentication Provider, SIP provides htpasswd, a program for configuring user logins. The resulting passwd file stores user names and encrypted passwords. For example:

```
operator:2TD4P8x16h38o
larry:C0uxYt.Ububrg
david:Z3EVMN/1mtcSc
```

As shipped with SIP, the passwd file will contain one user: ovuser with the password ovuser.

To use this authentication provider, you need to configure the portal users to be permitted to log in.

To Create User Logins and Passwords

• On UNIX:

As root, type:

/opt/OV/SIP/bin/htpasswd /opt/OV/SIP/etc/passwd <*username*>

• On Windows:

In a command window, type:

**"%SIP_HOME%\bin\htpasswd"
"%SIP_HOME%\etc\passwd" <*username*>**

where <*username*> is the name of the user you want to add. You will be prompted for the user's password.

NOTE   A word about htpasswd parameters:

You must use the default encryption mechanism. Do not use the -m and -s options to specify an encryption mechanism other than the default.

If you use -c to create a passwd file and the file already exists, you will overwrite the existing file and lose any entries in that file.

So that users need not share their passwords with you, the administrator, users can create their own password file entries with the -n option and mail them to you to add to the password file.

For your information, the location of the default password file is configured in the `SIPPath.properties` file located in the following directory:

Windows: `%SIP_HOME%\webapps\ovportal\`
UNIX: `/opt/OV/SIP/webapps/ovportal/`

To Delete User Logins

1. Using a text editor, open the `passwd` file located in the following directory:

   Windows: `%SIP_HOME%\SIP\etc\passwd`
   UNIX: `/opt/OV/SIP/etc/passwd`

2. Delete the login entry, and save the file.

LDAP Authentication Provider

The LDAP Authentication Provider works by getting the username and password from the login page and determining whether it can bind with the LDAP server using that username and password. It does not look up the user's password in the LDAP database.

Registering the LDAP Authentication Provider

To use this authentication provider, register it with SIP, as described in "Registering an Authentication Provider" on page 257. An example registration follows.

Example Registration of the LDAP Authentication Provider

```
<Authentication
    LoginPage="/ovportal/jsp/security/login_html.jsp"
    AuthenticationProviderClass=
        "com.hp.ov.portal.security.LDAPAuthenticationProvider"
    ShowLogoutButton="yes"
    LogoutPage="/ovportal/jsp/security/logout_html.jsp"/>
```

Configuring the LDAP Authentication Provider

1. Configure the LDAP Authentication Provider. It is in the following file:

   Windows:
   `%SIP_HOME%\conf\share\authentication\LDAP\LDAP.xml`

UNIX:
`/opt/OV/SIP/conf/share/authentication/LDAP/LDAP.xml`

NOTE   The LDAP.dtd file is found in the same directory as the `LDAP.xml` file. The attributes defined in these files are described in Table 9-1 below.

---

2. Test the LDAP Authentication Provider configuration to confirm that you configured the parameters correctly. Run the command:

Windows:
`%SIP_HOME%\bin\LDAP.bat username "password"`

UNIX: **`/opt/OV/SIP/bin/LDAP username "password"`**

You must enclose the password in double-quotes as shown if it contains non-alphanumeric characters.

This command outputs to the command window verbose tracing of each step in the LDAP authentication. Any failures should indicate what parameter is configured incorrectly. Be sure to test each of the following cases:

- No such user (should fail)
- Valid user, wrong password (should fail)
- Valid user, correct password (should succeed)

3. To make the configuration changes take effect, stop and restart the servlet engine. For instructions on doing so, refer to "Restarting the Servlet Engine" on page 291.

Table 9-1   Attributes of the LDAPServerConfig Element

Attributes	Description
baseDN	A required attribute that defines the context in which the LDAP Authentication Provider will search the LDAP database for the user to authenticate. `baseDN` is a string. For example, `baseDN="ou=Employees, o=acme.com"`

Table 9-1   **Attributes of the LDAPServerConfig Element**

Attributes	Description

uidAttr	The user ID attribute to search for. For example, if the LDAP.xml file contains:
	
	  uidAttr="uid"
	
	and the user submits to the login page the name:
	
	  "Wile_E_Coyote@acme.com"
	
	and baseDN is:
	
	  baseDN="ou=Employees, o=acme.com"
	
	Then, the LDAP Authentication Provider will look up the user with the search:
	
	  (ou=Employees, o=acme.com, uid=Wile_E_Coyote@acme.com)

LDAPServer	A required attribute that defines the LDAP server to which to authenticate. This should be the server's fully-qualified domain name. For example: LDAPServer="ldap.acme.com"
LDAPPort	A required attribute that defines the port at which the LDAP server is listening. If useSSL is "yes", this should be the port at which the LDAP server is accepting SSL connections. The standard port for LDAP is 389, and for LDAP over SSL is 636, but you should be sure to determine what port your server is using. For example: LDAPPort="636"
useSSL	Should be "yes" if the LDAP server performs authenticated binds using an SSL connection, and "no" if it authenticates passwords submitted in plain text and not over an SSL connection. For example: useSSL="yes"

Table 9-1   **Attributes of the LDAPServerConfig Element**

Attributes	Description
authType	The type of authentication used by the LDAP server. Valid values of authType are: none: Use no authentication (anonymous) simple: Use weak authentication (clear-text password) sasl_mech: A space-separated list of SASL mechanism names. SASL is specified in RFC 2222. The LDAP authentication provider has only been verified to support "simple" authentication, but over an SSL connection (useSSL="yes"), this is reasonably secure. For example: authType="simple"
searchFirst	If this is "yes", the LDAP Authentication Provider will search for username in the database before attempting to bind as that user. This is appropriate if the uidAttr is an alias, and the LDAP server requires the user to bind as the real Distinguished Name returned by the search, if successful. Performance will be better if this can be "no", since the database need only be accessed once. For example: searchFirst="yes"

Web Server Authentication Provider

This authentication provider requires no user name and password if the user has already been authenticated by the web server. The user is considered authenticated if `HTTPServletRequest.getRemoteUser()` returns non-null. This assumes that you have set up one of the various web server authentication mechanisms and that the portal user has satisfied that mechanism's web server authentication.

The Web Server Authentication Provider that comes with SIP is an external authentication provider. For it to work, you must use only the forms of authentication that are supported by BOTH the web server and the servlet engine. Look at the Table 9-2 to identify what forms are supported by both the Tomcat servlet engine (used by SIP) and by IIS and Apache.

Table 9-2   **Forms of Authentications Supported by Web Servers and the Tomcat Servlet Engine**

	IIS Web Server	Apache Web Server	Tomcat Servlet Engine
HTTP Basic	X	X	X
HTTP Form	X	X	X
HTTP Digest	X	X	
HTTPS Client	X	X	
Windows Challenge/ Response	X		X

If you can use one of the three forms that are common to the web server and servlet engine—HTTP Basic, HTTP Form, and Windows Challenge/Response—configure them and then the Web Server Authentication Provider will recognize whether a user has been

authenticated. The scope and duration of the user's authentication are defined by the Web Server. Configuring user logins is a web-server-specific task; refer to Apache and IIS documentation.

If you must use one of the forms of authentication that is supported by the web server but not by Tomcat, you may need to use a different servlet engine, or else write a custom authentication provider that accesses the web server directly instead of through the servlet engine. These alternatives are not explicitly supported by SIP but may be possible solutions in this situation.

To use this authentication provider, register it with SIP, as described in "Registering an Authentication Provider" on page 257. An example registration follows.

Example Registration of the Web Server Authentication Provider

```
<Authentication
  LoginPage="/ovportal/jsp/security/noAuthAvail_html.jsp"
  AuthenticationProviderClass=
  "com.hp.ov.portal.security.WebServerAuthenticationProvider"
  ShowLogoutButton="no"/>
```

NNM SSO Authentication Provider

This authentication provider requires no user name and password, if the user has already been authenticated by the NNM Session Manager web login mechanism. This authentication provider requires that NNM be running on the same host as SIP.

To use this authentication provider, register it with SIP, as described in "Registering an Authentication Provider" on page 257. An example registration follows.

Configuration for NNM SSO Authentication (See Note below)

```
<Authentication
  LoginPage=
   "http://<NNM_SIP_host>/OvCgi/ovlaunch.exe?URL=/ovportal/"
  AuthenticationProviderClass=
   "com.hp.ov.portal.security.NNMSSOAuthenticationProvider"
  ShowLogoutButton="yes"
  LogoutPage="/ovportal/jsp/security/logout_html.jsp"/>
```

NOTE

The NNM SSO Authentication Provider requires that NNM is running on the same host as SIP. The *<NNM_SIP_host>* referred to in Figure is the fully-qualified hostname of the machine running SIP and NNM.

On Windows, the NNM SSO Authentication example in Figure on page 255 will work without further configuration.

On UNIX, note that the login page URL is using the default port, rather than port 8880, which the NNM Web GUI normally uses on UNIX. If you want the NNM Web GUI to authenticate for SIP, it must run on the same web server (and same port) as SIP, or else the cookies the NNM Web UI creates will not be visible to SIP.

To make this work on UNIX, add the following to the SIP apache web server configuration file /opt/OV/SIP/apache/conf/httpd.conf:

```
Alias /OvDocs/ /opt/OV/www/htdocs/
ScriptAlias /OpenView /opt/OV/www/cgi-bin/OpenView
ScriptAlias /OvCgi/ /opt/OV/www/cgi-bin/
Alias /OvBackgrounds/ /opt/OV/share/backgrounds/
```

Registering an Authentication Provider

After you choose an authentication provider, you need to indicate to SIP the one to use. The authentication provider is a portal-wide setting configured in the OVPortalConfig.xml file.

1. Open OVPortalConfig.xml located in the following directory:

   Windows: %SIP_HOME%\conf\framework\
   UNIX: /opt/OV/SIP/conf/framework/

2. Find the Authentication element in the file, and edit the following attributes:

   - LoginPage (**required**)

   - AuthenticationProviderClass (**required**)

   - ShowLogoutButton

   - LogoutPage

   - SessionTimeout

   For sample configurations of each of the supplied authentication providers, see Figure through Figure . For an explanation of each of the Authentication attributes, see Table 9-3 on page 258.

   Definitions of the Authentication child element are located in Table 9-3 on page 258.

3. When finished, save the file.

4. After changing OVPortalConfig.xml, you must stop and restart the servlet engine before the changes take effect in SIP. You can do so through the SIP Administration Pages or from the command line. For instructions, see "Restarting the Servlet Engine" on page 291.

NOTE   If you have a SIP server specifically for other display types, such as PDA or WML, use the corresponding *_<devicetype>.jsp files; for example login_wml.jsp or login_pda.jsp. The examples in Figures 10-1 through Figure 10-6 refer to the HTML device type.

Table 9-3   **Configurable Attributes of the Authentication Element**

Attribute	Description
`LoginPage`	The required `LoginPage` attribute is the URL of the login page used by the Authentication Provider. Portal and External Authentication Providers use this URL differently.
	A Portal Authentication Provider requires a login page containing a form with an action that reinvokes ovportal with the parameters `J_USERNAME` (user name) and `J_PASSWORD` (password). For an example, see the login page supplied with SIP, located in:
	Windows: `%SIP_HOME%\webapps\ovportal\jsp\security\login_html.jsp`
	UNIX: `/opt/OV/SIP/webapps/ovportal/jsp/security/login_html.jsp`
	If the login page is on the same host as SIP, this is referred to with the relative URL: `/ovportal/jsp/security/login_html.jsp`
	An External Authentication Provider can use several different types of login pages, depending on the security policy it implements. For example, the Null Authentication Provider uses `/ovportal` as its login page, since all logins automatically succeed.
	The NNM SSO Authentication Provider uses the login page of the NNM Web UI: `http://localhost/OvCgi/ovlaunch.exe?URL=/ovportal`
	Note that in order for this to work, the NNM Web UI must share the same host and web server port as SIP, because otherwise the cookies created by ovsessionmgr will not be valid for the SIP session.
	The WebServerAuthenticationProvider uses the "No Authentication Available" error page as its login page, since either the web server has already taken care of authentication, or some misconfiguration has occurred.

Table 9-3   Configurable Attributes of the Authentication Element

Attribute	Description
`AuthenticationProviderClass`	The fully-qualified name of a Java class that implements either the `PortalAuthenticationProvider` interface or the `ExternalAuthenticationProvider` interface. For example: `AuthenticationProviderClass="com.hp.ov.portal.security.FileAuthenticationProvider"` Also see "Javadocs" on page 263.
`ShowLogoutButton`	If the Authentication Provider permits logout, configure `ShowLogoutButton="yes"`. For example, logout is meaningful in the case of the File Authentication Provider, but not in the case of the Web Server Authentication Provider (since you would have to exit the web browser to log out).

`LogoutPage`	If the authentication provider permits logout, and if it has information or controls that it wishes to present to the user after logout, configure the `LogoutPage` attribute with the URL to the logout page. For example:
	
	`LogoutPage="/ovportal/jsp/security/logout.jsp"`
	
	For an example, see the login page supplied with SIP, located in:
	
	Windows:
	`%SIP_HOME%\webapps\ovportal\jsp\security\logout_html.jsp`
	
	UNIX:
	`/opt/OV/SIP/webapps/ovportal/jsp/security/logout_html.jsp`

Table 9-3   **Configurable Attributes of the Authentication Element**

Attribute	Description
SessionTimeout	The time, in seconds, that a SIP session can remain inactive before SIP will automatically terminate the session. Automatic portal refreshes do not count as activity.
	The default session time out is 32400 seconds, or 9 hours. SessionTimeout="0" means use the application server's default session time out, normally 30 minutes. SessionTimeout="-1" means never time out.
	Consider carefully before setting SessionTimeout="-1". Resources associated with the sessions will not be released, which in turn may cause the application server's memory requirements to grow unacceptably large.

Developing a Custom Authentication Provider

If none of the supplied authentication providers meets your needs, you can write and integrate your own custom authentication provider. SIP provides APIs and sample code to write your own Java class to integrate with SIP. Following is a general idea of what is involved in this type of integration.

SIP Authentication Model

The SIP authentication model loads a configured Authentication Provider Java class. If one of the supplied Java classes is not sufficient, you can create a Java class that implements a custom authentication provider and integrate it with SIP.

In general terms, creating a custom authentication provider involves:

- Deciding the type of Authentication Provider to develop: Portal Authentication Provider or External Authentication Provider

- Developing the code

- Developing a login page

- Developing a logout page

- Making the authentication provider configurable, if relevant

- Registering the authentication provider with SIP

Decide Upon the Type of Authentication Provider

First, you need to decide whether you are going to develop a Portal Authentication Provider or an External Authentication Provider. Use the following general principles and supplied samples to guide your decision.

General Principles

You should implement the `PortalAuthenticationProvider` interface if the user's session (the "state" of whether the user is logged in) is held locally, in SIP.

You should implement the `ExternalAuthenticationProvider` interface if the user's session is held externally to SIP.

`PortalAuthenticationProvider` implements the `Authenticate` method, which accepts username and password as input and makes the decision whether password actually authenticates username.

`ExternalAuthenticationProvider` implements the `getAuthenticatedUser` method, which determines whether there is an authenticated user or not (determined through some means known only to itself), and if so, returns the username to SIP.

Supplied Authentication Providers As Examples

Use the description of the supplied authentication providers in Table 9-4 on page 262 as examples when developing a custom authentication provider.

Table 9-4   Supplied Authentication Providers

Supplied Authentication Providers	Type and Description
NullAuthenticationProvider	An External Authentication Provider, because the user's state of being logged in is external to SIP (always true) and because the user never inputs a username or password, and the Authentication Provider returns a username (always "anyuser") to SIP.
NoPasswordAuthenticationProvider	A Portal Authentication Provider, because the user's state of being logged in is local to SIP, and because it accepts a username from the user.
FileAuthenticationProvider	A Portal Authentication Provider, because the user's state of being logged in is local to SIP, and because it authenticates based on a username and password input by the user.

Table 9-4   Supplied Authentication Providers (Continued)

Supplied Authentication Providers	Type and Description
LDAPAuthenticationProvider	A Portal Authentication Provider, because the user's state of being logged in is local to SIP, and because it authenticates based on a username and password input by the user. It does perform an authenticated bind with the remote LDAP server, but the LDAP server does not maintain this bind for the duration of the session; the LDAP Authentication Provider merely delegates the password validation task to the LDAP server.
NNMSSOAuthenticationProvider	An External Authentication Provider, because the user's state of being logged in is external to SIP (held by ovsessionmgr, part of NNM). SIP never sees the username and password input by the user, and it returns the username (which it gets from a cookie) to SIP.
WebServerAuthenticationProvider	An External Authentication Provider, because the user's state of being logged in is external to SIP (held by the web server). SIP never sees the username and password input by the user, and it returns the username (which it gets by calling HttpServletRequest.getRemoteUser()) to SIP.

Develop the Code

Following is general information that will help you develop your custom authentication provider code.

Development Resources

You have several resources for the development of the authentication provider.

Javadocs

Javadoc for the Authentication Provider interfaces can be found at:

- Windows: `%SIP_HOME%\htdocs\javadoc\index.html`
- UNIX: `/opt/OV/SIP/htdocs/javadoc/index.html`

Sample Authentication Providers

Start with one of the two sample Authentication Provider java files:

- `SampleExternalAuthenticationProvider.java`
- `SamplePortalAuthenticationProvider.java`

Located in:

- *Windows:* `%SIP_HOME%\samples\authentication`
- *UNIX:* `/opt/OV/SIP/samples/authentication`

Main Program

For testing purposes, it can be helpful to add a main method to your Authentication Provider which outputs results to standard output.

For a PortalAuthenticationProvider, the main method would interpret its first two command line arguments as username and password, pass them to the Authenticate method, and output detailed tracing.

An ExternalAuthenticationProvider could be tested as a CGI program, so that it can access cookies.

ExternalAuthenticationProvider

An ExternalAuthenticationProvider must implement the `getAuthenticatedUserName` method. Refer to the javadoc for details.

Since an ExternalAuthenticationProvider maintains state with a remote authentication service, if it supports logout, it may need to clear this state when the user logs out. For example, it may need to remove the session cookie.

PortalAuthenticationProvider

A PortalAuthenticationProvider must implement the `authenticate` method. Refer to the javadoc for details.

The logout method of a PortalAuthenticationProvider normally does not need to do anything.

Developing a Login Page

The login page you develop is different based on whether you are developing an External Authentication Provider or a Portal Authentication Provider.

ExternalAuthentication Provider

An ExternalAuthenticationProvider may or may not use a login page. Two models are possible.

If the user has already logged in when the portal is first accessed, the getAuthenticatedUserName method can detect this and return the correct user name. If not, it returns null and the Portal redirects to the login page, which either is an error page ("You have not already logged in") or is the login page for the external authentication server, which gathers whatever parameters it requires and authenticates however it chooses. From the point of view of the SIP program, all that is required is that the next time the getAuthenticatedUserName method is called, it can return a user name. Normally this requires a session cookie to be set. See the Javadoc for the ExternalAuthenticationProvider interface for a detailed description of how to do this.

PortalAuthenticationProvider

The login page associated with a PortalAuthenticationProvider is required to forward to /ovportal/ with the query parameters J_USERNAME and J_PASSWORD set. Normally this would be accomplished with a form with "action=/ovportal/". The Portal calls the PortalAuthenticationProvider.authenticate method with these parameters to authenticate the user.

When developing your login page, start with /ovportal/jsp/security/login_html.jsp.

Located in:

Windows:
%SIP_HOME%\webapps\ovportal\jsp\security\login_html.jsp

UNIX:
/opt/OV/SIP/webapps/ovportal/jsp/security/login_html.jsp

This supplied login page also has the following useful features, which you can copy:

- If Javascript is enabled, it positions the cursor on the username field when the page is first displayed.

- If Javascript is enabled, typing the username and then hitting either tab or newline forwards the user to the password field.

- If Javascript is enabled, typing the password and then hitting newline submits the form.

- If Javascript is not enabled, the login page does work correctly, though it is not as user-friendly.

Developing a Logout Page

The portal redirects to the configured logout page when the user logs out. This page can issue a message, such as, "If you want to be sure none of your confidential data escapes, exit the browser now." Or it can permit the user to login again, with a button linking to /ovportal/.

When developing your logout page, start with /ovportal/jsp/security/logout_html.jsp.

Located in the following directory:

Windows: %SIP_HOME%\webapps\ovportal\jsp\security
UNIX: /opt/OV/SIP/webapps/ovportal/jsp/security

Make a Custom Authentication Provider Configurable (if Relevant)

Your authentication provider may have configuration parameters. See for example the sample configuration file for the LDAP Authentication Provider, located in:

Windows: %SIP_HOME%\conf\share\authentication\LDAP\LDAP.xml

UNIX: /opt/OV/SIP/conf/share/authentication/LDAP/LDAP.xml

Register a Custom Authentication Provider with SIP

You must register your authentication provider with SIP by configuring the Authentication element in the file.

Windows: %SIP_HOME%\conf\framework\OVPortalConfig.xml
UNIX: /opt/OV/SIP/conf/framework/OVPortalConfig.xml

For information on registering an authentication provider, see
"Registering an Authentication Provider" on page 257.

For a detailed description of the authentication elements, refer to
Table 9-3 on page 258.

10 Integrating SIP Into Your Own Environment

Configuring SIP in a Distributed Environment

Service Information Portal provides support for distributed and shared configuration files. User roles, user preferences, portal views, and module configuration files can be shared or distributed across a network.

Multiple SIP servers can share portal configuration, using HTTP or HTTPS to distribute and share SIP configuration. This makes it easier to maintain multiple SIP servers, because configuration changes only need to be made in one place instead of on each SIP server. The SIP configuration server can be in the DMZ or behind another firewall.

Distributed and Shared Configuration Files

Service Information Portal provides a mechanism for configuring remote access to these configuration files through the `SIPPath.properties` file. User roles, user preferences, portal views, and module configuration files can be shared or distributed across the network.

Only two entries in `SIPPath.properties` can be configured for remote access through HTTP or HTTPS: `SIP_SHARE_CONF_DIR` and `SIP_PASSWD_FILE`. The other directories must be local directories on the SIP server.

Configuring the SIP_Path.properties file

1. Using an ASCII editor, open the SIPPath.properties file:

   Windows: `%SIP_HOME%\webapps\ovportal\SIPPath.properties`

   UNIX: `/opt/OV/SIP/webapps/ovportal/SIPPath.properties`

2. Set `SIP_SHARE_CONF_DIR` to a URL or shared file system path where your shared SIP configuration is installed. The path must contain the equivalent of `/opt/OV/SIP/conf/share/` in a local SIP configuration.

Figure 10-1   **URL Example**

```
SIP_SHARE_CONF_DIR=http://<YourServer.YourCompany.com>/OvShareDocs/
```

You must configure the web server with an appropriate alias for
/OvShareDocs/. For example, you would configure an apache web server
by adding a line like the following to httpd.conf:

```
Alias /OvShareDocs/ /opt/OV/SIP/conf/share
```

NOTE   The OVShareDocs directory should contain the files that would otherwise
appear in the /share directory:

Windows: %SIP_HOME%\conf\share\
UNIX: /opt/OV/SIP/conf/share/

Figure 10-2   HTTPS URL Example

```
SIP_SHARE_CONF_DIR=https://<YourServer.YourCompany.com>/OvShareDocs/
```

You can also serve shared configuration from a secure web server. In
order for this to work, you must configure a Java SSL implementation,
like JSSE, on the SIP client systems. For documentation of how to install
JSSE, see the whitepaper *Configuring SIP to Use Secure Socket Layer
(SSL) Protocol for HTTPS.* You can navigate to the whitepaper at the
following web site: http://www.openview.hp.com.

Figure 10-3   UNIX Shared File Path Example

```
SIP_SHARE_CONF_DIR=/net/YourServer/opt/OV/SIP/conf/share
```

Clients must be UNIX systems. You can configure this using the NFS
automounter on the client system and exportfs(1M) on a HP-UX server
or share(1M) on a Solaris server.

The user that SIP is running as must have access to the remote files. By
default, the Tomcat process is configured to run under the user "www"
(on HP-UX) and "nobody" (on Solaris). A potential problem is that the
user may have a different user id on the SIP server than it does on the
remote system. For example, on the SIP system, user "www" has user id
101. On the remote system with the /conf/share directory, "www" has
user id 30. When the id is passed in the authentication check, the
authentication will fail. The solution is to make sure that "www" has the
same user id on both systems. Another possibility is running Tomcat as
some other new or existing user, provided that both systems agree on the
user_name/user_id mapping.

Figure 10-4   **Windows Shared File Path Example**

```
SIP_SHARE_CONF_DIR=file://YourServer/<ShareName>
```

This example is one of several ways to mount the remote directory. You could also use the Windows `Map Network Drive` dialog.

In the example in Figure 10-4, you share the folder `%SIP_HOME%\conf\share` using the *ShareName* on the server system. The SIP client, which must be Windows, automatically accesses the shared file over the network. The server can be a UNIX system if you have NFS on the Windows system.

By default, the Tomcat service is configured to run as the "system" account. When SIP tries to access shared resources on Windows, it only has the default local system account privileges, and may not have access to shared drives on other machines. In this situation, configure the Tomcat service to be executed as a user who has access to the remote directory. The user should have "read" and "change" privileges to the remote directory.

In the Windows `Services` window, select the `Tomcat` service and view the properties. Select the `Log On` tab and specify the username and password of the user that Tomcat should run under. Stop and restart the service, and SIP will have the privileges that are granted to the given user.

NOTE   When SIP is running in the default configuration (using the default system account), it may not have permission to access all remote resources.

Table 10-1   **Default Port and Protocol Settings for Shared SIP Configurations**

Server Product	Protocol	Port	Configuration
SIP	http	80	Protocol and port configurable by changing the value of `SIP_SHARE_CONF_DIR` in the `SIPPath.properties` file. The default is a local file system directory.
	https	443	

Whitepaper on Setting Up SIP in a Distributed Environment

For the whitepaper *Configuring Service Information Portal (SIP) to Work Within a Distributed Secure Environment*, go to `http://www.openview.hp.com` and navigate to the whitepapers.

Setting Up SIP to Use Secure Socket Layer (SSL) Protocol for HTTPS

The diagram in Figure 10-5 below shows the application server using HTTPS/SSL to serve SIP data to a client. It also shows the application server being served data using HTTPS/SSL from back-end servers, such as the LDAP server, SIP server, and any other server that can communicate via HTTPS.

Figure 10-5   **Communication using HTTPS/SSL**
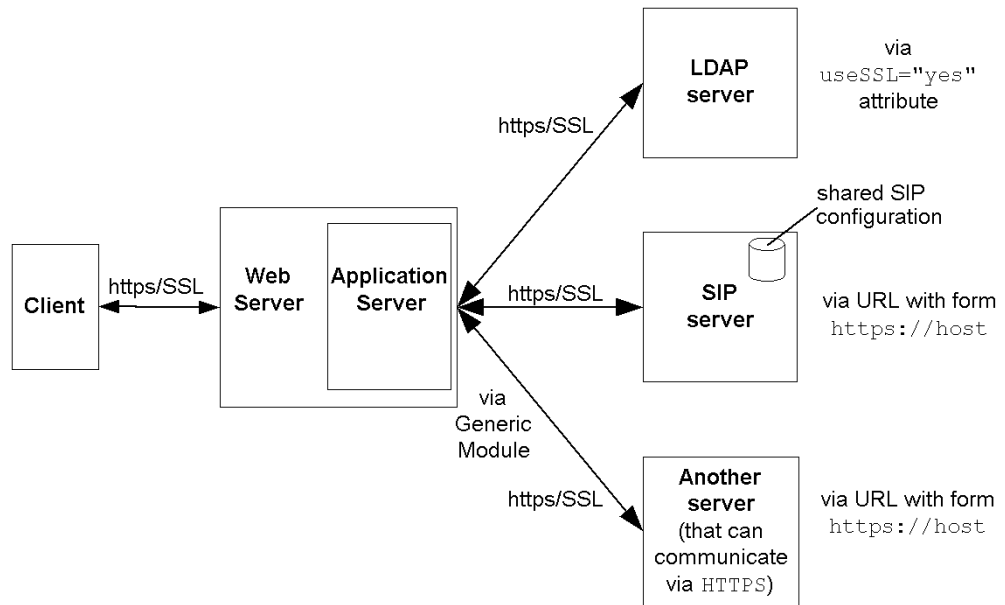
Enabling the Application Server to Serve SIP Data Using HTTPS

SIP uses Tomcat as its application server. Follow this procedure to enable access to SIP through HTTPS:

- Install SIP onto a host that has a secure web server installed, and configure SIP to run on the secure web server. For IIS, configure it as a secure web server.

- When accessing SIP from the client browser, use a URL of the form `https://<hostname>/ovportal`

Enabling SIP to Communicate with a Back-end Server Using HTTPS

You can enable SIP to be an HTTPS client and get its back-end server content through SSL. This requires that JSSE, or another package that enables Java programs to act as an HTTPS client, is installed on the application server.

Also, the application server must have certificates so that it trusts the secure web servers on the host systems. For installation instructions, reference the whitepaper *Configuring SIP to Use Secure Socket Layer (SSL) Protocol for HTTPS*. You can navigate to the whitepaper at the following web site: `http://www.openview.hp.com`.

- For LDAP, set the SIP attribute `useSSL="yes"` in the following file:

  Windows:
  `%SIP_HOME%\conf\share\authentication\LDAP\LDAP.xml`

  UNIX:
  `/opt/OV/SIP/conf/share/authentication/LDAP/LDAP.xml`

  For more information, see "LDAP Authentication Provider" on page 250.

- For other hosts that can communicate via HTTPS, use a URL of the form `https://host`

Enabling HTTPS in the SIP Configuration Editor

The option to select HTTPS when configuring OVIS and OVR through the `SIP Configuration Editor` is disabled, unless you have configured SIP to communicate through SSL. Check the product documentation that came with your version of the OVIS or OVR management software to verify that HTTPS is fully supported. Then reference the whitepaper *Configuring SIP to Use Secure Socket Layer (SSL) Protocol for HTTPS*. You can navigate to the whitepaper at the following web site: `http://www.openview.hp.com`.

Configuring SIP to Run in a Wireless Environment

You can use several of the SIP modules to display data on web-ready cell phones and personal digital assistants (PDAs) that support a web browser:

• NNM Network Device Health and Alarms modules

• OVO Messages module

• Service Health and Service Cards modules

• Internet Services module

• Bookmarks and Message Board modules

• Generic modules

Recommended Approach to Using SIP in a Wireless Environment

The optimal way to run SIP in a wireless environment is to have a second SIP installation that serves wireless clients only. To run SIP on both cell phones and PDAs, you may even want two separate SIP installations for these.

• SIP works best when a single display type (such as HTML) is configured per SIP server. If multiple client types access the same SIP server, problems can arise, particularly with WML clients.

• For authentication providers and login and logout pages, SIP cannot select a page that is appropriate for the client. One authentication mechanism can be configured per SIP installation. If you are using the same authentication provider and login page for your regular SIP environment and the wireless environment, you can use one SIP installation. If not, you will need a separate installation of SIP for the wireless environment.

How SIP Works in the Wireless Environment

SIP associates each type of client with a particular display type: web browsers are associated with HTML, cell phones are associated with WML, and personal digital assistants are associated with PDA. Based upon the display type, SIP selects the correct Java Server Pages (JSPs) and XSL style sheets. When a module displays output, the module selects a XSL document which also uses the display type, and in so doing, presents module information in a way that is best suited for the client.

The mapping of display types is made through the webapps/ovportal/saagents.xml file. This file is configured by default for HTML, PDA, and WML-capable devices. The file contains mappings of client names (for example, Netscape), specific client identification strings (for example, Mozilla/4.6), and display types (for example, HTML).

Configuration of SIP for the wireless environment is done through direct editing of XML files. Table 10-2 on page 279 shows all files related to running SIP in a wireless environment. Table 10-3 on page 280 lists and describes elements and attributes of the saagents.xml file.

Configuring SIP to Run Wireless

Assuming that you have chosen to run your wireless SIP on its own SIP server, you have a few configuration issues to be aware of and tasks to do.

Differences Between Standard Deployment of SIP and Wireless Deployment

There are several differences between how SIP behaves in a regular environment versus a wireless environment:

- Users cannot select (or switch) roles. The initial (default) role is always used.

- Users do not logout of SIP when running wireless. Instead, they just disconnect from the portal.

- Tabs and columns are not presented. All modules are displayed in the order in which they appear in the portal view file.

- Users cannot edit the modules or portal views.

Client-Specific Markup Issues

The Message Board, Bookmarks, and Generic modules display or reference data sources that may contain client-specific information. For example, the Generic module may reference a web page or site that supports HTML 4.0. This content may not be displayable in a PDA.

While WML contains markup that may be identical to HTML and displayable in Netscape or Internet Explorer, the reverse is rarely true. (Only certain tags can be displayed in both WML and HTML: paragraph, bold, italic, and table tags.) In addition, WML clients require well-formed XML syntax while web browsers usually do not.

For these reasons, the content displayed by the modules may produce errors when displayed through some clients. There is no method for filtering the display of non-compliant modules for a given display type. The only solution for certain clients is to not use the module at all.

Other Data Issues Specific to Individual Clients

Links for a HTML device usually do not work for a WML device. Also, you may run into a problem if you are trying to display too much data through the device.

Changing the Defaults in the saagents.xml File

Change the `saagents.xml` file to specify as the default the type of client the SIP server needs to support. The file is found in:

Windows: `%SIP_HOME%\webapps\ovportals\saagents.xml`
UNIX: `opt/OV/SIP/webapps/ovportal/saagents.xml`

For information on the elements and attributes of the `saagents.xml` file, see Table 10-3 on page 280.

Creating Portal Views Specific to Cell Phones and PDAs

Based on sample files supplied with SIP, you will create special portal views that display well in the wireless environment. The samples are can be found in:

Windows:
`%SIP_HOME%\conf\share\views\samples\mobileDemo_pda.xml`
`%SIP_HOME%\conf\share\views\samples\mobileDemo_wml.xml`

UNIX:
`/opt/OV/SIP/conf/share/views/samples/mobileDemo_pda.xml`
`/opt/OV/SIP/conf/share/views/samples/mobileDemo_wml.xml`

Tips for Creating Portal Views

- Only the portal view that is associated with the user's initial role is displayed.

- Portal tabs do not work in a wireless display. If modules in the portal view file are organized on tabs, they will be listed one after another, Place your most important modules at the beginning of the list to avoid lengthy scrolling.

Starting SIP in a Wireless Client

Start SIP by typing: `http://<server_name>/ovportal`

Log in as a valid SIP user.

Listing of XML Files and Elements

Configuration of SIP for the wireless environment is done through direct editing of XML files. Table 10-2 on page 279 shows all files related to running SIP in a wireless environment. Table 10-3 on page 280 lists and describes elements and attributes of the `saagents.xml` file.

Table 10-2   Files Related to Running SIP in a Wireless Environment

File and Location	Purpose
Windows: `%SIP_HOME%\webapps\ovportals\saagents.xml` *UNIX:* `/opt/OV/SIP/webapps/ovportal/saagents.xml`	Contains mappings of client devices to the display type.
Windows: `%SIP_HOME%\conf\framework\OVPortalConfig.xml` *UNIX:* `/opt/OV/SIP/conf/framework/OVPortalConfig.xml`	File in which SIP authentication provider (including a login page) is configured.

Table 10-2   Files Related to Running SIP in a Wireless Environment

File and Location	Purpose

Windows: `%SIP_HOME%\webapps\ovportal\jsp\security`
UNIX: `/opt/OV/SIP/webapps/ovportal/jsp/security`
 	
`login_pdf.jsp`
`login_wml.jsp`
 	
`logout_pdf.jsp`
`logout_wml.jsp`
 	
`noAuthAvail_pdf.jsp`
`noAuthAvail_wml.jsp`
 	
`refresh_pdf.jsp`
`refresh_wml.jsp`
 	
`userName_pdf.jsp`
`userName_wml.jsp`
	Directory where JSPs are stored.

Windows: `%SIP_HOME%\conf\styles` *UNIX:* `/opt/OV/SIP/conf/styles`	XSL files.

Windows:
`%SIP_HOME%\conf\share\views\samples\`
 	
UNIX: `/opt/OV/SIP/conf/share/views/samples`
 	
`mobileDemo_pda.xml`
`mobileDemo_wml.xml`
	Sample wireless-specific portal view files.

Table 10-3   Mappings in the saagents.xml File

XML Elements and Attributes	Description
`agent device-name`	This attribute is the canonical client name. It is not used in SIP 3.2, however, it provides an easy way to identify each name when reading the `saagents.xml` file.
`display-type`	This attribute is a canonical name that best represents the client. This string is used in the JSP and XSL file names, so it should be a single word consisting of ASCII alphanumeric characters. The values recognized by SIP are "`html`", "`pda`", and "`wml`".

Table 10-3   **Mappings in the saagents.xml File (Continued)**

XML Elements and Attributes	Description
user-agent	This element is the client ID string. It represents a sub-string of the client's user-agent HTTP request field. The ID string may match any or all of this field. When the portal is looking for a match, it takes the first user-agent that matches the request field, so list the most specific user-agent first in the file.

Running SIP in Non-English Language Mode

Following are required configuration tasks that help you prepare to run SIP on a system and browser that have been configured to operate in non-English language mode.

Configuring the Browser Settings

SIP supports display of characters in UTF-8 code set to the Web browser. Make sure your web browser is either in the "Auto-Select" or "UTF-8" modes to ensure that the data is properly displayed.

If the Web browser has "Auto-Select" enabled for Encodings, the UTF-8 data should be displayed correctly through the portal. If not, you will need to set the browser's Encoding to "Unicode (UTF-8)".

Configuring Japanese Fonts (HP-UX 11.x only)

On HP-UX 11.x, in order to display Japanese fonts properly, you must do the following:

1. Make sure the HP-UX Java JDK 1.3.1_01 (which contains the Ricoh fonts) is installed. You can download it from the HP web site:
   `http://www.hp.com/java`

2. Make sure two HP-UX 11 patches are installed: PHSS_24931 and PHSS_25091.

3. Make a backup of `fonts.properties.ja`, and then link the `fonts.properties.ja.Ricoh` file to `fonts.properties.ja`:

   ```
   cd /opt/java1.3/jre/lib
   mv fonts.properties.ja fonts.properties.ja.ORIG
   ln -s fonts.properties.ja.Ricoh fonts.properties.ja
   ```

4. For the SIP software, edit `/etc/rc.config.d/ovsip` (which contains SIP's environment settings) and add the following lines:

```
JAVA_FONTS=/usr/lib/X11/fonts/TrueType/japanese.st/typefaces
Export JAVA_FONTS
```

5. For the `SIP Configuration Editor`, manually set `JAVA_FONTS` and `LANG` in your environment.

6. Set the `DISPLAY` variable to an active X server session used by SIP to the system where the Japanese fonts are available.

NOTE   You may need to adjust the settings in `/etc/rc.config.d/ovsip` to make sure the `DISPLAY` is pointing at the local system's X server (or alternatively, the Virtual Networking Computer (VNC) server running on the SIP server) upon startup of SIP.

7. Restart the servlet engine to enable the changes made to `/etc/rc.config.d/ovsip`. For instructions, see "Restarting the Servlet Engine" on page 291.

Configuring the XML Files Used By SIP

If non-ASCII characters are added to XML, XSL, or JSP files, you must preserve the UTF-8 codeset for these characters.

NOTE   The Microsoft Windows Notepad editor allows files to be saved in the UTF-8 code set. On many UNIX platforms, the `iconv` command can be used to translate from Shift JIS (or any other code set) into UTF-8 to make editing in a non-UTF-8 codeset simpler.

Scaling SIP: Increasing the Amount of System Memory Used

By adjusting the JVM memory configuration to more system memory, you can improve how SIP runs in large environments.

Adjusting JVM Memory Usage Values for the Tomcat Servlet Engine

Most Java Virtual Machines (JVMs) are limited to a heap size of 65 Mb by default. Some command line options are available to set the maximum size of the JVM's memory, as well as the initial size: -Xms (initial size) and -Xmx (maximum size).

On Windows

1. Using the Windows registry editor, go to:

```
HKEY_LOCALMACHINE\SOFTWARE\Apache Software Foundation\Tomcat Service
Manager\Tomcat\Parameters
```

2. Open the JavaOptions registry key for editing.

   The JavaOptions registry key is a text string containing values for several parameters. The last two parameters, Xms and Xmx, set the minimum and maximum memory usage for the JVM. Edit the value of these parameters as follows:

   -Xms256M

   -Xmx512M

3. Set the value of the JavaOptions registry key and close the Registry Editor.

On UNIX

The following script can be edited to change the JVM parameters:

/etc/rc.config.d/ovsip

JVM Default Parameters:

INITIAL_HEAP_SIZE=256m
MAX_HEAP_SIZE=512m

If you change these parameters, make sure you follow the correct syntax. The values of the JVM parameters must be in powers of 2 (for example, 32, 64, 128, 256, and so forth) and followed by the shortened form of "megabyte", which is "m".

Adjusting JVM Memory Usage Values for SIP Scripts

You can modify other SIP scripts to adjust the default memory usage values. Change the values of `INITIAL_HEAP_SIZE=32m` and `MAX_HEAP_SIZE=128m` to reflect the desired memory configuration.

Windows:
```
$SIP/bin/create_role_db.bat
$SIP/bin/SIPConfig.vbs
```

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

11 Performing Routine Administrative Tasks

Setting Portal Logging and Tracing

A portal log file is available to help you troubleshoot problems with the portal. The log file contains warning and error entries:

- A warning is logged to indicate that a problem occurred, but the SIP framework or a module was able to continue, possibly using a default value or some assumption. The warning is issued because the assumption may have been wrong and you might want to fix the problem.

- An error is logged when a problem prevents the SIP framework or a module from completing a task.

Portal tracing is primarily intended for use by HP support. It supplies a large amount of extra data that may be used to debug portal problems. You are advised to turn on portal tracing only if instructed to do so by support personnel. For errors, read the error description for instructions on fixing it.

Viewing Log Files

1. Log in as a user who has access to a special SIP Administrator role. For more information, "Understanding Special SIP Administrator Roles" on page 79.

2. Switch to the SIP Administrator role, if it is not already displayed.

3. Click the SIP General Admin tab.

4. To see entries that are of "warning" and "error" severity, go to the Logging and Tracing segment and click the View log file [View] button. In the case of warnings, check the assumption made by the program. Fix the configuration, if needed.

NOTE   Only the last 500 kb of a very large log file is displayed in the SIP user interface. If you want to examine the contents of the entire file, see the following:

Windows: %SIP_HOME%\log\sip.log
UNIX: /opt/OV/SIP/log/sip.log

Setting Trace Level and Viewing Trace Files

Turn on portal tracing only if instructed to do so by support personnel.

1. Log in as a user who has access to a special Administrator role. For more information, "Understanding Special SIP Administrator Roles" on page 79.

2. Switch to the `Administrator` role, if it is not already displayed.

3. Click the `SIP General Admin` tab.

4. In the `Logging and Tracing` segment of the page, set the trace level.

   Tracing can be turned on for all SIP sessions on the server or for the current session only.

   Setting the trace level to anything above "none" causes entries to be written to the portal trace file. If you set it to "warning," it will have the same contents as the log file.

5. Click [Apply] at the bottom of the portal page.

6. Display the trace file by clicking the View trace file [View] button.

   A trace level of "tracing" results in entries that record key steps during the portal's execution.

   A trace level of "verbose" does the same, but with greater detail.

NOTE   If you change the tracing level and go to another tab before clicking [Apply], your trace setting is not applied.

NOTE   Only the last 500 kb of a very large trace file is displayed in the SIP user interface. If you want to examine the contents of the entire file, see the following:

Windows: %SIP_HOME%\log\sip.trace
UNIX: /opt/OV/SIP/log/sip.trace

Increasing the Size of the Log File

The default size for a log file is approximately 10 MB. When that size is reached, the log file is copied to `sip.log.old` or `sip.trace.old`, and a new file is started. It is possible to lose information if a file called `sip.<tracelog>.old` already exists.

You can modify the maximum log size by directly editing the `maxLogSize` value in `OVPortalConfig.xml` located in:

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

Controlling the Servlet Engine

SIP is a Java servlet that is accessed through the Tomcat servlet engine. SIP is restarted by stopping and restarting the servlet engine.

Restarting the Servlet Engine

After making certain configuration changes, you must restart the servlet engine before changes take effect:

- After adding or changing a module registration file.

- After making changes to the authentication provider configuration.

- In other situations where you are specifically instructed to do so.

To Restart the Servlet Engine from the SIP Administration Pages

Be aware that you and all other SIP users will be logged out when you restart the servlet engine.

1. Log in as a user who has access to a special SIP Administrator role. For more information, see "Understanding Special SIP Administrator Roles" in the *SIP Deployment and Integration Guide* (SIP_Deployment_Integration.pdf).

2. Switch to the SIP Administrator role, if it is not already displayed.

3. Click the SIP General Admin tab.

4. In the Servlet Engine Control segment, click [Restart].

To Restart the Servlet Engine from Outside of SIP

Windows:
From the Control Panel, select Services. Stop and then restart Tomcat. Alternatively, you can use the command line: **net stop tomcat** and **net start tomcat**.

UNIX:
As root, stop and restart the web server and servlet engine by running the following. (The DISPLAY variable must be configured prior to restarting the web server and servlet engine, unless DISPLAY is set in /etc/rc.config.d/ovsip.)

Stop on HP-UX: `/sbin/init.d/ovsip stop`
Start on HP-UX: `/sbin/init.d/ovsip start`

Stop on Solaris: `/etc/init.d/ovsip stop`
Start on Solaris: `/etc/init.d/ovsip start`

Running Garbage Collector

SIP can force the JVM to attempt to free up any previously requested system memory that is currently not being used.

1. Log in as a user who has access to a special SIP Administrator role. For more information, "Understanding Special SIP Administrator Roles" on page 79.

2. Switch to the `SIP Administrator` role, if it is not already displayed.

3. Click the `SIP General Admin` tab.

4. In the `Servlet Engine Control` segment, click `[Garbage Collect]`.

Determining the Amount of Time It Takes to Display Each Module

If you are having performance problems with a portal, you can determine the amount of time each module takes to load. Using the Timer URL parameter, you can display the time, in milliseconds, at the bottom of each module.

Turning On the Timer

- In the browser window, enter the following:

http://<*yourmachinename*>/ovportal?Timer=yes (or `true`)

Turning Off the Timer

- In the browser window, enter the following:

http://<*yourmachinename*>/ovportal?Timer=no (or `false`)

Displaying More Detailed Error Messages in the Portal Interface

Through the portal you may see short error messages, such as "Data Not Configured" and so forth. You can display more verbose error information by turning on a URL parameter.

Turning On Portal Verbosity

• In the browser window, enter the following:

`http://<yourmachinename>/ovportal?portalVerbosity=true`

Turning Off Portal Verbosity

• In the browser window, enter the following:

`http://<yourmachinename>/ovportal?portalVerbosity=false`

Improving Performance

Certain SIP modules require significant initial load times because a large amount of data is passed over the network when the module is first accessed. As a result, the first person displaying these modules experiences the longest delay. To enhance your customer's experience, log into the portal after SIP has been restarted and open any portal view that contains the following modules (if used in your environment):

- NNM Alarms module

- Network Device Health module

- Topology Map module

- One of the following modules:

    — Service Graph

    — Service Browser

    — Service Health

    — Service Card

If you open it first, the required information is already cached when your customers access their portals. Your customers won't experience the delay. It is not necessary to perform a log in for each portal user. A single log in, viewing the above listed modules, is sufficient.

NOTE

If a tab page containing one of the NNM modules remains blank with the progress bar partially loaded, one of the NNM management stations (that SIP gathers data from) may be in the early phase of an NNM backup procedure.

Wait for the NNM backup to proceed beyond the ovpause state. The modules display when the NNM management station issues an ovresume command. If the browser timeout limit is exceeded while you are waiting, you must press [Refresh] to display the modules.

For more information, see the Troubleshooting section of the *NNM Integration with SIP* manual (NNM_Integration.pdf).

Monitoring the Size of the NNM snmpCollect Database

If automatic configuration of NNM SNMP data collection has been enabled, periodic trimming of the NNM SNMP data collector database will be necessary. For details, see "Monitoring the Size of NNM's snmpCollect Database" in the *NNM Integration with SIP* manual (`NNM_Integration.pdf`).

Checking the Contents of the Role Database

After running create_role_db, you can run the test_role_db
command to verify the contents of the created role database.

- To find out the configured default user and default role, run:
  test_role_db

- To see the database entry for a particular user, run:
  test_role_db -u *<user>*

- To see the database entry for a particular role, run:
  test_role_db -r *<role*

A   Understanding SIP Configuration Files and XML

What You Need to Know About SIP Configuration Files and XML

SIP uses XML as the means of storing and retrieving all configuration information. XML is used to represent configuration data on the Service Information Portal server. XML is not being sent to the web browser over the wire. It is being used to store configuration files on the file system.

This appendix explains how SIP uses XML for various types of configuration information.

Use of XML for SIP Configuration Files

SIP configuration files can be categorized according to nine configuration types. Table A-1 lists the types of configurations and the associated file or directory. The remaining sections in this appendix describe how each type of configuration is configured.

You can also understand the configuration files from a visual standpoint by looking at the SIP directory structure diagrams in the following GIF files:

```
dirStructureSIP3a.gif
dirStructureSIP3b.gif
dirStructureSIP3c.gif
```

The files are in the following directory:

Windows: `%SIP_HOME%\htdocs\C\help\SIP\`
UNIX: `/opt/OV/SIP/htdocs/C/help/SIP/`

Table A-1   **SIP XML Configuration Files**

Configuration Type	File or Directory
Portal configuration	*Windows:* `%SIP_HOME%\conf\framework\OVPortalConfig.xml` *UNIX:* `/opt/OV/SIP/conf/framework/OVPortalConfig.xml`
Module registration	*Windows:* `%SIP_HOME%\SIP\registration\` *UNIX:* `/opt/OV/SIP/registration/`
Module configurations	*Windows:* `%SIP_HOME%\conf\share\modules\` *UNIX:* `/opt/OV/SIP/conf/share/modules/`
Management station configurations	*Windows:* `%SIP_HOME%\conf\share\stations\` *UNIX:* `/opt/OV/SIP/conf/share/stations/`
LDAP authentication configuration	*Windows:* `%SIP_HOME%\conf\share\authentication\LDAP\LDAP.xml` *UNIX:* `/opt/OV/SIP/conf/share/authentication/LDAP/LDAP.xml`

Table A-1   **SIP XML Configuration Files (Continued)**

Configuration Type	File or Directory
Customer model configuration	*Windows:* %SIP_HOME%\conf\share\organizations\ *UNIX:* /opt/OV/SIP/conf/share/organizations/
User Role configuration	*Windows:* %SIP_HOME%\conf\share\roles\ *UNIX:* /opt/OV/SIP/conf/share/roles/
Portal view configuration	*Windows:* %SIP_HOME%\conf\share\views\ *UNIX:* /opt/OV/SIP/conf/share/views/
User preferences configuration	*Windows:* %SIP_HOME%\conf\share\users\ *UNIX:* /opt/OV/SIP/conf/share/users/

Portal Configuration

The Portal Configuration file—OVPortalConfig.xml—stores global settings that apply to all portals. (Note, however, that the portal header, portal footer, and help topic can be overridden at the portal view level.) Here are the configurations stored in the OVPortalConfig.xml file:

- Tracing level (configured through the SIP Administration Pages).

- Maximum log size (configured through direct editing of OVPortalConfig.xml).

- Names of the portal header and portal footer JSP files (configured through direct editing of OVPortalConfig.xml).

- Authentication provider configuration (configured through the SIP Administration Pages).

- List of skins display through the [Options] button on the SIP button bar (configured through direct editing of OVPortalConfig.xml).

- List of refresh rates available through the [Options] button on the SIP button bar (configured through direct editing of OVPortalConfig.xml).

- Customer model sources configuration (configured through the SIP Administration Pages).

- Customer model export destinations configuration (configured through the `SIP Administration Pages`).

Module Registration

Module registration files indicate to the SIP what capabilities the module provides. A registration file defines information, such as a unique ID for the module, a reference to the servlet, the module's associated default help file, and the capabilities of the module.

Module Configuration

Some modules can be configured more extensively than others: some offer global configurations that affect every instance of a particular module; some offer settings you can make to individual module instances; and some offer the ability to configure display filters. All module configurations are specified in XML files.

For instructions on configuring module-specific settings, see the SIP manuals that explain the integrations of individual management products: "Getting Additional Documentation" on page 17.

LDAP Authentication Configuration

If you decide to use the LDAP authentication provider, you need to configure the LDAP configuration file supplied with SIP. In this configuration file, you specify the LDAP server and port, as well as other LDAP-specific information.

Customer Model Configuration

The SIP customer model configuration is made up of multiple registered customer model sources that are either XML files or programs that generate XML. The XML must be based on the SIP Simple Customer Model (SCM), which maps *organizations* to *nodes*, *interfaces*, and *services*, and is defined in an XML DTD.

User Role Configuration

SIP uses an authorization model called the User Role model. By associating users with roles and defining for each role what you want the user to be able to see and do, you achieve portal security. The user role

configuration defines all valid portal users, roles, and management data filters and the associations among them. Most configuration tasks are performed through the SIP Configuration Editor. Only the advanced configurations are done through direct editing of the XML.

Portal View Configuration

A portal view configuration is a configured set of modules and how they appear on tabs. It is also a configured set of portal view attributes, such as name in the portal button bar, skin (portal look and feel), portal refresh rate, default tab, portal header, portal footer, and others.

Portal view configurations are based on several DTDs, for the purpose of modularity:

- The overall DTD—PortalView.dtd—defines the overall structure of the portal page.

- The filter.dtd describes the filtering elements.

- The remaining DTDs are specific to each module that is configured in the portal view file: For example, OVAlarms.dtd, OVBookmarks.dtd, OVGeneric.dtd, etc.

  Sample View Files — Supplied sample view files provide a starting point for the creation of new ones. Several samples are provided with SIP.

User Preferences Configuration

Users who are associated with a role that has the editing permissions level of "UserPreferences" or "ViewAdmin" can customize the name that appears in the portal button bar and the skin in which the portal is displayed. Because these two attributes are set in a portal view file that can be associated with multiple users, when a user makes a change to one or both of these attributes, a user preferences configuration is created for the user. The XML file is named for the user's login.

Changes made to User Options are stored as <*login*>.xml in the User Preferences directory.

B Refresh Model for SIP Configuration Data

Refreshing Modified SIP Configuration and Data Files

Depending upon the type of configuration or data file, changes to those files take effect in different ways:

- Some changes take effect when you **access** the information by displaying or refreshing the portal page.

- Some changes take effect on a scheduled, **periodic** basis, based on a configurable timer that determines the refresh rate for the information.

- Some changes take effect when you perform an administrative action, such as a command, that **forces** information to be refreshed.

- Some changes take effect when you **restart** the servlet engine. Certain data is loaded when the portal servlet is first initialized and is not refreshed unless the servlet engine is stopped and restarted.

Table B-1 on page 307 lists each refresh method described above, and indicates the type of configuration or data that uses each method. It also indicates how to refresh the data or file.

Table B-1   **Refresh Model for SIP Configuration and Data Files**

Refresh Method	Configuration or Data	How to Refresh the Data or File
Access	Password file LDAP authentication data Portal views User preferences NNM Alarms data NNM Topology data OVO data OVSN data OVIS data OVR data XSL style sheets Java Server Pages (except for `common.jsp`)	Display or refresh the portal page.

Table B-1   Refresh Model for SIP Configuration and Data Files (Continued)

Refresh Method	Configuration or Data	How to Refresh the Data or File
Periodic	Customer Model	Changes to the Customer Model take effect on a periodic basis, depending upon the value of the `timeout` attribute in the `OVPortalConfig.xml` file. This attribute can be set on the `Customer Model` tab in the `SIP Administration Pages`.
	Object capabilities data from the NNM object database (`ovwdb`) used for filtering	Changes take effect on a periodic basis, depending upon the value of the `ovwdbTimeout` attribute of the `CustomerModelSources` element in the `OVPortalConfig.xml` file. This attribute can be set on the `Customer Model` tab in the `SIP Administration Pages`.
	NNM Data Collector data	The data used by the Network Device Health module gauges is gathered by NNM according to the schedule (in minutes) specified in the `rawDataRefresh` attribute in the `netHealthConfig.xml` file. The NNM data collection configuration, itself, is updated to match current SIP requirements by manual command line entry (`ovcolautoconf`) or according to a schedule set up by the administrator
	NNM Symbol registration information	If changes are made within NNM to the "Symbol Type" assigned to particular devices, SIP receives the changes according to the schedule established by the `symbolFetchRateInMin` attribute in the `topologyConfig.xml` file.

Table B-1   **Refresh Model for SIP Configuration and Data Files (Continued)**

Refresh Method	Configuration or Data	How to Refresh the Data or File
Force	User Role Model	Changes to the User Role model take effect after you perform a successful `Save` operation in the `SIP Configuration Editor`, or rebuild the roles database using the following command: *Windows:* `%SIP_HOME%\bin\create_role_db.bat` *UNIX:* `/opt/OV/SIP/bin/create_role_db`
	Customer Model	`[Refresh]` button on the `Customer Model` tab in the `SIP Administration Pages`.

Restart	Module registration
	
	Module configuration
	
	Authentication provider configuration
	
	LDAP authentication configuration
	
	`SIPPath.properties`
	
	`common.jsp`
		Stop and restart the servlet engine. For instructions, see "Restarting the Servlet Engine" on page 291.

C   Rules for Direct Editing of XML Files

Rules for Direct Editing of XML Files

- Make a backup before modifying XML files.

- Understand editing permissions on XML files.

- Validate the XML after you modify it.

- Be careful not to lose changes made through the GUI. This can happen when you edit through the XML file and edit through the GUI at the same time.

Backing Up XML Files

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

Understanding Editing Permission on XML Files

When using the editing windows within the SIP portal, the web server needs to have read/write permissions to the underlying files in order to save your changes. The apache web server and SIP run as:

Solaris: user `"nobody"`

HP-UX: user `"www"`

At runtime, `umask` is set by the `tomcat.sh` script to 022, so files are created mode 0644 and directories created mode 0755.

Therefore, at install time, SIP sets permissions and ownership for files to `mode 0644` and directories to `mode 0755`. If you add or change anything, make sure directories are owned by the appropriate user specified above, files set to `mode 0644`, and directories set to `mode 0755`.

For `tomcat` to operate properly, the following directories and all files underneath them need to have the correct permissions set (user as specified above, files set to mode 0644, and directories are set to mode 0755):

- `/opt/OV/SIP/tomcat`
  (directory only, so tomcat can create the work directory when needed)

- `/opt/OV/SIP/tomcat/conf`
  (directory only)

- `/opt/OV/SIP/tomcat/logs`
  (directory, all subdirectories, and all files)

- `/opt/OV/SIP/tomcat/webapps`
  (directory, all subdirectories, and all files)

- `/opt/OV/SIP/tomcat/work`
  (directory, all subdirectories, and all files)

For SIP to operate properly, these directories and all `.xml` files (not `.dtd` files) underneath them need to have the correct permissions set (`user` set to anyone with editing permissions, files set to mode 0644, and directories are set to mode 0755):

- `/opt/OV/SIP/conf/share/organizations`
  (directory, all subdirectories, and all .xml files)

- `/opt/OV/SIP/conf/share/users`
  (directory, all subdirectories, and all .xml files)

- `/opt/OV/SIP/conf/share/modules`
  (directory, all subdirectories, and all .xml files)

- `/opt/OV/SIP/conf/share/roles`
  (directory, all subdirectories, and all .xml files)

- `/opt/OV/SIP/conf/share/views`
  (directory, all subdirectories, and all .xml files)

Validating XML Files

The Service Information Portal will detect and report an invalid XML configuration file. However, after you make modifications to XML files, you may want to validate your XML syntax.

Provided with SIP is the command `xmlvalidate`, which checks whether the XML file is both well-formed and valid. This command uses the same XML parser as SIP, so if the file passes `xmlvalidate`, it will work with SIP.

For the command to work from outside the `bin` directory, add the following to your PATH variable:

Windows: `%SIP_HOME\bin`
UNIX: `/opt/OV/SIP/bin`

For XML files that reference a DTD file residing in the same directory, correct usage of the `xmlvalidate` command is:

```
xmlvalidate -v <xml filename>
```

An XML file is "well-formed" if it conforms to a minimal set of rules defined for all XML documents. It is "valid" if it conforms to the DTD listed at the beginning of the XML file.

If a file does not reference a DTD or if its DTD is unavailable, `xmlvalidate` can be used to determine if the file is well-formed without checking for validity as defined by the DTD. The form of this command is:

```
xmlvalidate -V <xml filename>
```

Sometimes an error reported by xmlvalidate may not clearly indicate how to fix the problem. For example, a message like "Attribute 'name' must be declared for element type 'XYZ', is an indication that the attribute 'name' may have been misspelled.

As an alternative to xmlvalidate, you can find an XML validation tool for Windows NT at `www.xmlspy.com`.

Avoiding Loss of Changes

If you are using the portal interface to change a configuration and directly editing the XML configuration file at the same time, be careful not to lose the changes made through the interface by writing out the file over the interface changes.

D   Default SIP Portal and Protocol Settings

Port and Protocol Defaults for SIP Framework, Customer Model, and Modules

If you plan to install multiple management products on the same machine, you need to be aware of the default protocols and ports used by product. Following are three tables that list the default port and protocol settings for:

- SIP Framework
- Customer Model
- SIP Modules

Table D-1   **SIP Framework Default Port and Protocol Settings**

SIP Component	Server Product	Protocol	Port	Configuration
SIP Shared Configuration	SIP	http	80	Protocol and port configurable by changing the value of `SIP_SHARE_CONF_DIR` in `SIPPath.properties` file. The default is a local file system directory.
		https	443	
LDAP Authentication	LDAP Server	ldap	389	Use of SSL and the LDAP server port can be configured through the file `conf/share/authentication/LDAP/LDAP.xml`
		ldap/ssl	636	

Table D-2   **Customer Model Default Port and Protocol Settings**

SIP Component	Server Product	Protocol	Port	Configuration
Customer Views Import	Customer Views	http	8880/UNIX 80/Windows	Server and port configurable through the URL specified as customer model source in the `Customer Model` tab of the `SIP Administration Pages`. E.g., `http://server:8880/OvCgi/ getcvdata.exe`
NNM Object Data Import	NNM 6.1	ovwdb	9999	NNM servers and `ovwdb` ports configurable through `conf/share/modules/NM/NNM Data.xml`
	NNM 6.2 or later	ovwdb	2447	
NNM Data Warehouse Export	NNM	http	8880/UNIX for NNM 6.3 and earlier 3443/UNIX for NNM 6.31 and later 80/Windows	Server and port configurable through the URL specified as customer model export destination in the `Customer Model` tab of the `SIP Administration Pages`. E.g., `http://server:8880/OvCgi/ ovsipexport.exe`
OV Reporter Export	OVR 4.0	http	80	Server and port configurable through the URL specified as customer model export destination in the `Customer Model` tab of the `SIP Administration Pages`. E.g., `http://server/HPOV_IOPS/c gi-bin/repimport.exe`

Table D-3   **SIP Modules Default Port and Protocol Settings**

SIP Component	Server Product	Protocol	Port	Configuration
Alarms	NNM 6.1	ovalarmsrv	2345	Use `SIP Configuration Editor` to configure ports (`OVAlarmSrv Port` on `NNM` tab for management stations).
	NNM 6.2 or later	ovalarmsrv	2953	
Network Device Health	NNM	http	8880/UNIX for NNM 6.3 and earlier 3443/UNIX for NNM 6.31 and later 80/Windows	Use `SIP Configuration Editor` to configure port (`Web Server Port` on `NNM` tab for management stations).
Topology	NNM	ovw	3700 to 3700+n	n=highest OVW session number
	NNM 6.1	ovwdb	9999	Use `SIP Configuration Editor` to configure ports (`OVwDB Port` on `NNM` tab for management stations).
	NNM 6.2 or later	ovwdb	2447	
	NNM	http	8880/UNIX for NNM 6.3 and earlier 3443/UNIX for NNM 6.31 and later 80/Windows	Use `SIP Configuration Editor` to configure ports (`Web Server Port` on `NNM` tab for management stations).
Internet Services	OVIS 3.5, 4.0	http	80	Use `SIP Configuration Editor` to configure protocol and port (`OVIS` tab for management stations).

Table D-3   SIP Modules Default Port and Protocol Settings (Continued)

SIP Component	Server Product	Protocol	Port	Configuration
OVO Messages	OVO for UNIX	OVO DB	1521	Use `SIP Configuration Editor` to configure ports (`OVO` tab for management stations).
	OVO for Windows	http	80	
		https	443	
	OVSN	opsvcterm	7278	
Service Browser, Service Cards, Service Graph, Service Health, Custom Service View	OVSN	opsvcterm	7278	Use `SIP Configuration Editor` to configure ports (`OVO` tab for management stations).
Service Desk 4.0, Service Desk 4.0 for Specialists, Service Desk 4.5	OVSD	http	80	Use `SIP Configuration Editor` to configure Service Desk server through the Role Property `ServiceDesk.SSPserver`. Web server port can be configured through the module default XML file or the module instance.
NNM Reports, OVO Reports, OVR Performance Reports	OVR	http	80	Use `SIP Configuration Editor` to configure protocol and port (`Reporting Station` tab for management stations).

Table D-3   SIP Modules Default Port and Protocol Settings (Continued)

SIP Component	Server Product	Protocol	Port	Configuration
Bookmarks	Various	http	80	Server, port, and protocol configurable through the URL specified through the `Bookmarks` tab of the `SIP Administration Pages` or the Bookmarks module edit page accessed from the module title bar.
		https	443	
Message Board	SIP	file system		
Generic Module	Various	http	80	Server, port, and protocol configurable through the URL specified as the `href` for the module default XML file or the module instance.
		https	443	

E Configuring Basic Features of the User Role Model Using XML

Configuring the User Role Model Through XML

The user role files are located in the following directory:

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

Creating a User Role Package and Adding It to the User Role Model

1. In the `roles` directory, create a user role package file by copying and renaming the supplied `package.xml` file. This file is supplied as a starting point for new user role packages and is located in the following directory:

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

2. Add the new package to the User Role model by adding an entry for it in the following file:

   Windows: `%SIP_HOME%\conf\share\roles\index.xml`
   UNIX: `/opt/OV/SIP/conf/share/roles/index.xml`

   For example:

   `<UserRolePackageRef href="<new_package>.xml"/>`

3. Save and close the `index.xml` file.

4. In the package file, define the new roles, referring to the following instructions and to the `Role` element information in Table E-5 on page 331.

Creating Roles

Roles are defined in user role package configuration files. These files contain some set of users and roles and together define all the users and roles that make up the User Role model. All users and roles can be defined in a single file, or they can be partitioned into multiple files (for example, one for each customer or organization).

The package files define the UserRolePackage element shown in
Figure E-2 and described in Table E-3. You can name the package files
anything you want as long as you reference them in the index file.

Choosing the Portal View for a Role

A portal view defines what someone in a given role is allowed to see
through the portal.

Assuming that you have already created portal views for the roles that
you are now creating, determine which portal view you want to associate
with each role. Portal view files are stored in the following directory:

Windows: %SIP_HOME%\conf\share\views\
UNIX: /opt/OV/SIP/conf/share/views/

If you have not yet created portal views, see Chapter 6, "Customizing
Portal Views," on page 111.

For a description of the PortalViewRef element and attributes, see
Table E-7 on page 332.

Configuring a Role So Changes Are Not Made to an Original Portal View File

Refer to the copy attribute of the PortalViewRef element in Table E-7
on page 332.

Choosing the Level of Edit Permissions for a Role

See Table E-8 on page 335.

Overriding the Edit Permissions Level

If you want to change the edit permissions level for a user but you do not
want to create a new role, you can override the edit permissions defined
on the role itself. For more information, see the EditPermissions
element in Table E-6 on page 332.

Defining Management Data for a Role

Before you define the management data for a role, you need to verify that
the customer and their associated resources are defined in the Customer
Model. In the OVPortalConfig.xml file, the CustomerModel hrefs

indicate the names and locations of the customer model sources. Look in those files and other customer model sources and verify that the customer (organization) and resources are defined.

Table E-1   **Examples of Valid MgmtData Element Values**

Data to Be Presented	
All Data	All known data should be presented; no data should be filtered out. This is a `MgmtData` element that has no `orgName` attribute and no `OrganizationFilter` child element. `<MgmtData/>`

No Data	No data should be presented. This is a `MgmtData` element that has no `orgName` attribute and an empty `OrganizationFilter` child element.
	
	`<MgmtData>`
	`  <OrganizationFilter/>`
	`</MgmtData>`

Data for a Single Organization	Data for a single organization (or customer) should be presented. This is a `MgmtData` element that has an `orgName` attribute and no `OrganizationFilter` child element. `<MgmtData orgName="Acme"/>`

Table E-1   **Examples of Valid MgmtData Element Values (Continued)**

Data to Be Presented	
Data for Multiple Organizations	Data for multiple organizations (or customers) should be presented. This is a MgmtData element that has a non-empty OrganizationFilter child element.

```
<MgmtData name="GoldCustomers">
    <OrganizationFilter>
        <OrganizationRef href="Acme"/>
        <OrganizationRef href="Nabob"/>
        <OrganizationRef href="Aureum"/>
    </OrganizationFilter>
</MgmtData>
<MgmtData name="GoldCustomers">
    <OrganizationFilter>
        <OrganizationRef href="Acme"/>
        <OrganizationRef href="Nabob"/>
        <OrganizationRef href="Aureum"/>
    </OrganizationFilter>
</MgmtData>
<MgmtData name="GoldCustomers">
    <OrganizationFilter>
        <OrganizationRef href="Acme"/>
        <OrganizationRef href="Nabob"/>
        <OrganizationRef href="Aureum"/>
    </OrganizationFilter>
</MgmtData>
<MgmtData name="GoldCustomers">
    <OrganizationFilter>
        <OrganizationRef href="Acme"/>
        <OrganizationRef href="Nabob"/>
        <OrganizationRef href="Aureum"/>
    </OrganizationFilter>
</MgmtData>
<MgmtData name="GoldCustomers">
  <OrganizationFilter>
    <OrganizationRef href="Acme"/>
    <OrganizationRef href="Nabob"/>
    <OrganizationRef href="Aureum"/>
  </OrganizationFilter>
</MgmtData>
```

Setting Role Properties

Refer to "Setting Role Properties" on page 326 and "Giving Your Module Access to SIP Data Through Variable Substitution" on page 180.

Creating Users

Like roles, users are defined in user role package configuration files. Again, these files contain some set of users and roles and together define all the users and roles that make up the User Role model. All users and roles can be defined in a single file, or they can be partitioned into multiple files (for example, one for each customer or organization).

To Create Users

In the `roles` directory, open the user role package file in which you want to define the users. This file is somewhere under the following directory:

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

5. In the package file, define the new users, referring to the following instructions and to the `User` element information in Table E-4 on page 330. For example:

```
<User name="operator" displayName="Operator"
initialRole="NOC"/>
```

End the user XML definition with either "`/>`" or `</User>`.

NOTE   Each user must have an `initialRole`. This attribute is mandatory and specifies the role that the user will be in when he or she first logs in to SIP.

Assigning Roles to Users

After you have defined the roles, you can assign to each user which roles (in addition to the `defaultRole`) you want them to have.

• Use the `RoleRef` element and `href` and `title` attributes to associate a role with a given user. For example:

```
<User name="operator" displayName="Operator"
      initialRole="NOC">
  <RoleRef href="NOC"/>

  <RoleRef href="AcmeTechnical" title="Acme Technical"/>
  <RoleRef href="AcmeBusiness" title="Acme Business"/>
</User>
```

The title attribute is only needed if you want to override the title defined for the role.

Assigning Multiple Roles Using Wildcards

The RoleRef href can contain wildcards using Perl5 pattern matching. This allows multiple roles to be associated with the user using a single href. All roles that match the pattern specified by the href are added to the list of available roles for the user. For example, an href value of ".*" matches all roles. An href value of "Acme.*" matches all roles that start with "Acme".

If a RoleRef with wildcards has an EditPermissions child element, the user gets the specified edit permissions for all roles that match the pattern.

User Role Model Elements and Attributes

The user role DTD is located in the following directory:

Windows: %SIP_HOME%\conf\share\roles\UserRole.dtd
UNIX: /opt/OV/SIP/conf/share/roles/UserRole.dtd

Figure E-1   Sample index.xml File

```
<UserRoleModel>
    <UserRolePackageRef href="default.xml"/>
    <UserRolePackageRef href="samples.xml"/>
</UserRoleModel>
```

Figure E-2   Sample package.xml file

```
<UserRolePackage title="Sample Users and Roles">
  <!-- Users -->
  <User name="admin" displayName="Admin" initialRole="Welcome">
    <RoleRef href="Welcome"/>
    <RoleRef href="LiveDemo"/>
    <RoleRef href="AcmeTechnical" title="Acme Technical">
```

```
      <EditPermission level="ViewAdmin"/></RoleRef>
  </User>
  <!-- Roles -->
  <Role name="Welcome" title="Welcome">
    <PortalViewRef href="welcome.xml"/>
    <EditPermission level="UserPreferences"/>
    <MgmtDataRef href="NoData"/>
  </Role>

  <Role name="LiveDemo" title="Live Demo">
    <PortalViewRef href="samples/liveDemo.xml"/>
    <EditPermission level="ViewAdmin"/>
    <MgmtDataRef href="AllData"/>
  </Role>
  <Role name="AcmeTechnical" title="Technical">
    <PortalViewRef href="samples/technical.xml"
      copy="Acme/technical.xml"/>
    <EditPermission level="UserPreferences"/>
    <MgmtDataRef href="AcmeOrg"/>
  </Role>
<!-- Management Data -->
<!-- This will show data only for the "Acme" organization -->
  <MgmtData name="AcmeOrg">
    <OrganizationFilter>
      <OrganizationRef href="Acme"/>
    </OrganizationFilter>
  </MgmtData>
```

NOTE   Sample user role package files can be found in the following two
                 directories:

                 Windows:
                 %SIP_HOME%\conf\share\roles\
                 %SIP_HOME%\samples\authorization\roles\

                 UNIX:
                 /opt/OV/SIP/conf/share/roles/
                 /opt/OV/SIP/samples/authorization/roles/

Table E-2   User Role Model Elements and Attributes

Elements and Attributes	Description
UserRoleModel	The root element of the `/conf/roles/index.xml` file. This file is an index to all packages that together make up the User Role model. Each package is stored in a separate XML file. A User Role model consists of one or more packages.
UserRolePackageRef	The child element of the `UserRoleModel` element. Each `UserRolePackageRef` element refers to a `UserRolePackage`.
href	An attribute of the `UserRolePackageRef` element. Specifies the file containing the user role package information. The value of `href` is interpreted as a filename relative to the `conf/share/roles` directory. It can include subdirectories. For example, the `href` value `gold/Apex.xml` is interpreted as the file `conf/share/roles/gold/Apex.xml`. The file so specified must be an XML document with `UserRolePackage` as the root element.

Table E-3   UserRolePackage Element

Elements and Attributes	Description
UserRolePackage	The root element of each user role package configuration file. Contains a set of user and role definitions. It can also contain management data definitions. A package can, for example, define all the users and roles for a particular customer or organization. Users in one package can reference `Role` and `MgmtData` definitions in another package.
title	An attribute of the `UserRolePackage` element. Optional descriptive string that describes the purpose of the package.
defaultMgmtData	An attribute of the `UserRolePackage` element. A reference to a `MgmtData` element. Specifies the management data to use for any role defined in the package that has a management data value of `DefaultMgmtData`. Makes it easy to specify the management data for all roles in the package in one place.

Table E-4   User Element

Elements and Attributes	Description
User	A child element of `UserRolePackage` element. Defines a valid SIP user. `User` element has `RoleRef` child elements. This list specifies the set of roles for the user. The order in which the `RoleRef` elements appear for a user determine the order in which the roles appear in the role drop-down menu in the user interface.
name	An attribute of `User`, name is a unique identifier that corresponds to the login name by which the user was authenticated to SIP.
displayName	An attribute of `User`, `displayName` is the name of the user that will be displayed in the portal welcome banner in the user interface, unless it is overridden in the user preferences.
initialRole	An attribute of `User`, initialRole is a mandatory attribute that specifies the role that the user will be in when he or she first logs in to SIP. It is a reference to a Role element. Because this is a required attribute, user must be defined as having at least one role.

If the "initial role" does not also appear in the list of roles for the user, it is implicitly added to the end of the roles for that user. |
| defaultUser | An attribute of `User`, `defaultUser` is a flag indicating that the user entry should be used as the "default user." This means that when a user logs in with a valid login name and there is no `User` entry for that login name and there is no portal view file named `conf/share/views/`*login*`.xml`, the user has all roles defined for the default user. |

Table E-5   Role Element

Elements and Attributes	Description
Role	A child element of `UserRolePackage` element. Defines a role. A user can have multiple roles, and multiple users can share a role. The `Role` element has four child elements that together define what a user operating in the role can see and do. Child elements: • `PortalViewRef` • `EditPermission` • `MgmtData`, `MgmtDataRef`, or `DefaultMgmtData` • `Properties`
name	An attribute of `Role`, `name` is a unique identifier for the `Role`. The name value is used in the `href` attribute of the `RoleRef` element to refer to this role. Only one role with a given name is allowed in the User Role model. Subsequent roles with the same name are ignored. The name must be unique across all user role packages.
title	An attribute of `Role`, title is a descriptive name for the role. The `title` is the string that appears in the drop-down list of roles presented to this user in the user interface. The title for a role does not need to be unique.
defaultRole	An attribute of `Role`, `defaultRole` is a flag indicating that the role should be used as the "default role." This means that when a user logs in with a valid login name and there is no `User` entry for that login name and there is a portal view file named conf/share/views/*login*.xml, that view is displayed to the user and the edit permissions and management data are determined by the default role. When this role is used as the default role (because it can also be used as a normal role), the `PortalViewRef` child element is ignored. Only the first `Role` element is recognized as the default role; any subsequent `defaultRole` flags are ignored.

Table E-6   RoleRef Element

Elements and Attributes	Description
RoleRef	A child element of the Role element. Allows multiple users to refer to the same role. The referenced role can be in any of the user role package files.
href	An attribute of RoleRef, href is a reference to a Role element. It must be the same as the name attribute of a Role element.
	Can contain wildcards using Perl5 pattern matching. This allows multiple roles to be associated with the user using a single href. All roles that match the pattern specified by the href are added to the list of available roles for the user. For example, an href value of ".*" matches all roles. An href value of "Acme.*" matches all roles that start with "Acme".
	If a RoleRef with wildcards has an EditPermissions child element, the user gets the specified edit permissions for all roles that match the pattern.
title	An optional attribute of RoleRef, title can be used to override the displayed role name as seen by the current user.
	If the title is specified for a RoleRef that contains wildcards, it serves as a prefix for the title of each of the roles that the regular expression matches.
EditPermissions	An optional child element of RoleRef. Allows a particular user to override the edit permissions defined on the role itself.

Table E-7   PortalViewRef Element

Elements and Attributes	Description
PortalViewRef	A child element of the Role element. References the portal view for this role. The portal view determines the set of tabs and modules that the user sees.

Table E-7   PortalViewRef Element (Continued)

Elements and Attributes	Description
href	An attribute of `PortalViewRef`, href specifies the file containing the `PortalView` element. The value of `href` is interpreted as a filename relative to the `conf/share/views` directory. It can include subdirectories. For example, the `href` value "`NetOperator.xml`" is interpreted as the file "`conf/share/views/NetOperator.xml`. The file so specified must be an XML document with `PortalView` as the root element.

Table E-7   PortalViewRef Element (Continued)

Elements and Attributes	Description
copy	An optional attribute of `PortalViewRef`, `copy` allows multiple roles to share a portal view file, and yet have a new copy of the portal view created if the portal view is modified (through GUI editing) through one of the roles.
	For example, you could set up one `NetOperator.xml` portal view file and have 50 roles (one each for 50 customers) share the file. Any changes made in `NetOperator.xml` would be seen by all 50 roles/customers.
	However, you could use the `copy` attribute to give editing permissions to five of these roles/customers in such a way that changes do not affect the original view file. Note that edit permissions are defined through the level. The "`copy`" attribute just determines where the changes go. When each of these five customers edits the portal view, it would be saved as a new portal view file under the file name specified by the `copy` attribute.
	Thus, the `copy` attribute specifies the filename to use to save a modified version of the original portal view. The value of `copy` is interpreted as a filename relative to the `conf/share/views` directory. It can include subdirectories. For example, the `href` value "`Apex/NetOperator.xml`" is interpreted as the file "`conf/share/views/Apex/NetOperator.xml`. The file specified by "copy" may or may not exist. SIP will create it as needed.
	Once the "copy" version of the portal view has been created, the role will use that file as the view file for the role (that is, it will no longer use the portal view file specified by the `href` attribute).
	Note that the `create_role_db` command creates view directories required by "copy" view files if they do not already exist. In the example just cited, it would create the directory conf/share/views/Apex.

Table E-8   EditPermissions Element

Elements and Attributes	Description
EditPermissions	A child element of the Role element. Specifies the editing operations the user can perform through SIP.
level	An attribute of EditPermissions, level specifies the level of editing operations permitted. Each level consists of an implicit set of operations. The levels are ordered. Each level allows all operations defined by the previous level. The three levels are: • ReadOnly • UserPreferences • ViewAdmin For an explanation of each level see "Understanding the User Role Model" on page 78.

Table E-9   MgmtData Element

Elements and Attributes	Description
MgmtData	A child element of UserRolePackage element. Specifies what management data will be presented when the associated role is used. The MgmtData element can appear directly with a Role, or it can be defined independently and referenced by multiple roles. The MgmtData element is essentially a list of organizations (or customers) as defined in the customer model in the OVPortalConfig.xml file. This list of organizations is translated at run-time into a list of resources (services, nodes, interfaces) that determine what information is presented to the user. For examples of valid MgmtData element values, see Table E-1 on page 324.

Table E-9   MgmtData Element (Continued)

Elements and Attributes	Description
name	An attribute of `MgmtData`, `name` is the unique identifier for the `MgmtData`. This is the value used in the `href` attribute of the `MgmtDataRef` element to refer to this management data specification.

Only one management data specification with a given name is allowed in the User Role model. If a subsequent one is found with the same name, it is ignored. |
| OrganizationFilter | An optional child element of the `MgmtData` element. Allows multiple organizations (or customers) to be specified for the "management data." |
| orgName | An attribute of `MgmtData`, `orgName` specifies the name of an organization (or customer) to use to determine the management data to present. This is provided as a shortcut so that a `OrganizationFilter` element does not need to be defined if there is only a single organization for the "management data" definition.

If both the orgName attribute and a child `OrganizationFilter` is specified, the organization specified by "`orgName`" is added to the list of organizations listed in the `OrganizationFilter`. |
MgmtDataRef	A child element of a `Role` element. MgmtDataRef is a way of specifying the management data for a role. It allows multiple roles to refer to the same "management data." The referenced "management data" can be in defined in any of the user role package files.
href	An attribute of `MgmtDataRef`, `href` is a reference to a `MgmtData` element. It must be the same as the `name` attribute of a `MgmtData` element.
DefaultMgmtData	A child element of the `Role` element. An indication that the role should use the `MgmtData` element specified by the `defaultMgmtData` attribute of the `UserRolePackage` element. This element is used to make this "inheritance" explicit.

Table E-10   **Properties Element**

Elements and Attributes	Description
Properties	An optional child element of the Role element. A list of Property elements.
Property	A child element of the Properties element. Specifies a role property name and value that can be used by modules, including the generic module.
name	An attribute of Property, specifies the name of the role property.
value	An attribute of Property, specifies the value of the role property.

F Configuring the Message Board and Bookmarks Modules Using XML

Creating Messages Through Direct Editing of XML

When directly editing XML files, see "Rules for Direct Editing of XML Files" on page 312. If you need information on the XML elements, see "Message Board DTD" on page 344 and "Portal View DTD" on page 142.

1. Create an HTML file with the contents of the message.

   The name of the HTML file must be unique, contain no spaces, and contain no non-alpha-numeric special characters. Also, the file name cannot begin with a forward slash (/) or with two periods followed by a forward slash (../).

2. Put the file in the `messageboard` directory or in a sub-directory of the `messageboard` directory:

   Windows: `%SIP_HOME%\conf\share\modules\messageboard\`
   UNIX: `/opt/OV/SIP/conf/share/modules/messageboard/`

3. Open the `messageDisplayNames.xml` file also located in the `messageboard` directory, and add a `Message` element that defines a title (`title` attribute) for the newly defined message file (`file` attribute), as shown in the following example:

   ```
   <MessageDisplayNames>
   <Message file="default" title="Default Message Board"/>
   <Message file="newMessage.html" title="What's New at
   Acme"/>
   </MessageDisplayNames>
   ```

   If your message is in a sub-directory of the `messageboard` directory, specify the file path relative to the `messageboard` directory, e.g., `file="subdir/mymessage"`.

   If you need information on the XML elements, see "Message Display Names DTD" on page 341.

NOTE   Notice that the file name can include the file extension or not.

4. When you are finished editing it, save and close the file.

5. Once the message files are in the messageboard directory, you can use the web interface to add, remove, rearrange them in the portal view file. Do so by clicking the [Edit] button on the Message Board module title bar.

Message Display Names DTD

The messageDisplayNames.xml file contains mappings between message file names and the title that will be displayed in the portal. messageDisplayNames.xml is also used as an index of available messages. The entries in this file become the list of Available Messages in the Message Board Edit GUI that is accessible when the user has ViewAdmin permissions.

The DTD is reproduced below. A complete example of a MessageDisplayNames element follows:

```
<MessageDisplayNames>
  <Message file="default" title="Default Message Board"/>
  <Message file="newMessage.html" title="What's New at Acme"/>
</MessageDisplayNames>
```

The DTD is stored in:

Windows: %SIP_HOME%\conf\share\modules\messageboard\
UNIX: /opt/OV/SIP/conf/share/modules/messageboard/

Figure F-1   messageDisplayNames.dtd

```
<!-- messageDisplayNames.dtd -->
<!-- Copyright (c) 2001 Hewlett-Packard Company -->
<!-- $Revision: /main/BACCHUS/1 $ -->
<!-- $Date: 2001/01/16 20:25 UTC $ -->
<!ELEMENT MessageDisplayNames (Message*)>
<!ELEMENT Message EMPTY>
```

```
<!ATTLIST Message
    file CDATA #REQUIRED
    title CDATA #REQUIRED>
```

Table F-1   **Message Attributes**

Attribute	Description
file	A required attribute of Message, specifies the name of the file that contains the message content and that is located in the messageboard directory. The name of file must match "filename" in the portal view XML Message element.
title	A required attribute of Message, specifies the message title that should be displayed in the portal interface.

Editing XML to Select Messages to Display in a Portal View

To add or remove messages that are displayed in a portal, you must edit the associated portal view file. Portal view files are located in or below the following directory:

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

When directly editing XML files, see "Rules for Direct Editing of XML Files" on page 312. If you need information on the XML elements, see "Message Board DTD" on page 344 and "Portal View DTD" on page 142.

Assigning Messages to a Portal View Using XML

1. Using an ASCII editor, open the portal view file that you want to display a particular message through.

2. Find the `MessageBoard` element, and edit it to add or delete the messages you want to display. Following is an example of a Message Board module instance with two messages defined for the `MessageBoard` element:

```
<ModuleInstance
    classid="com.hp.ov.portal.modules.messageboard"
    help="/OvSipDocs/C/help/SIP/messageView.html"
    id="module9" rollupState="down" title="Message Board">
  <MessageBoard>
    <Message filename="default"/>
    <Message filename="Welcome"/>
  </MessageBoard>
</ModuleInstance>
```

Messages are displayed in the order in which they appear in this file. "`filename`" is the only attribute of `Message`. It specifies the name of the file that contains the message content. The file is located in the `messageboard` directory.

3. Save and close the portal view file.

4. Log in as a user who has access to the appropriate role. Verify that the message was added.

Removing Messages from a Portal View Using XML

1. Using an ASCII editor, open the portal view file from which you want to remove a particular message.

2. Find the MessageBoard element, and delete the message filename of the message you no longer want to display through this portal view.

3. Save and close the portal view file.

4. Select a role that displays this portal view, and verify that the message was removed.

Message Board DTD

The OVMessageBoard.dtd file is used to define elements in the portal view file. The DTD is reproduced below. A complete example of the MessageBoard module in a portal view file follows:

```
<MessageBoard>
    <Message filename="default"/>
    <Message filename="newMessage.html"/>
</MessageBoard>
```

Figure F-2   **OVMessageBoard.dtd**

```
<!-- OVMessageBoard.dtd -->
<!-- Copyright (c) 2000 Hewlett-Packard Company -->
<!-- $Revision: /main/BACCHUS/1 $ -->
<!-- $Date: 2001/11/07 00:56 UTC $ -->

<!ELEMENT MessageBoard (Message)+>

<!ATTLIST MessageBoard
    id CDATA #IMPLIED>

<!ELEMENT Message EMPTY>

<!ATTLIST Message
    filename CDATA #REQUIRED>
```

Creating Groups of Shared Bookmarks Through Direct Editing of XML

There are two parts to setting up a group of shared bookmarks: creating the shared group entry in the sharedBookmarks.xml file, and adding a reference to the shared group into portal view files.

1. Using an ASCII editor, open the sharedBookmarks.xml file located in the following directory:

   Windows: %SIP_HOME%\conf\share\modules\bookmarks\
   UNIX: /opt/OV/SIP/conf/share/modules/bookmarks/

2. Add a SharedGroup element with a Group child element, as shown in the following example, which shows two groups being shared:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE SharedBookmarks SYSTEM "sharedBookmarks.dtd">
<SharedBookmarks>
   <Group name="hplist" title="HP Links (Shared Group)">
      <Entry title="HP OpenView Home"
            href="http://www.openview.hp.com"
            target="hpwin"/>
      <Entry title="HP Home" href="http://www.hp.com"
             target="hpwin"/>
   </Group>
   <Group name="group1" title="Group 1 Title">
      <Entry title="entry11" href="href1"
            target="group1win"/>
      <Entry title="entry12" href="href2"
            target="group1win"/>
   </Group>
</SharedBookmarks>
```

NOTE   The value of Group name is what is referenced from a portal view file as a way of designating this group as able to be shared.

3. Save and close the sharedBookmarks.xml file.

4. Open the portal view file to which you want to add a shared bookmark entry.

5. Find the `Bookmarks` module and add the `SharedGroup` entry. Define the name string, which is used to look up the group in the `sharedBookmarks.xml` file. See the example below:

```
<Bookmarks>
  <Group title="Generic Net Bookmarks">
    <Entry href="/OvSipDocs/C/demo/operations/Op_Trouble_Shooting.htm"
           target="bookmarkwin" title="Operations Troubleshooting"/>
    <Entry href="http://www.hp.com/e-services" target="bookmarkwin"
           title="Generic Net Primary Site"/>
  </Group>
  <SharedGroup name="hplist"/>
</Bookmarks>
```

6. Save and close the portal view file.

Shared Bookmarks DTD

The `sharedBookmarks.dtd` file is reproduced below and explained in Table F-3 on page 347 and Table F-2 on page 347. It is stored in the following directory:

Windows: `%SIP_HOME%\conf\share\modules\bookmarks\`
UNIX: `/opt/OV/SIP/conf/share/modules/bookmarks/`

Figure F-3   sharedBookmarks.dtd

```
<!-- sharedBookmarks.dtd -->
<!-- Copyright (c) 2001 Hewlett-Packard Company -->
<!-- $Revision: /main/BACCHUS/2 $ -->
<!-- $Date: 2001/01/30 22:23 UTC $ -->
<!ELEMENT SharedBookmarks (Group*)>
<!ELEMENT Group (Entry*)>
<!ATTLIST Group name CDATA #REQUIRED
      title CDATA #REQUIRED>
<!ELEMENT Entry EMPTY>
<!ATTLIST Entry title CDATA #REQUIRED
      target CDATA "_self"
      href CDATA #REQUIRED>
```

Table F-2   Attribute of the Group Element

Attribute	Description
name	A required attribute, name refers to the name of a shared group as referenced in the portal view file.

Table F-3   Attributes of the Entry Element

Attribute	Description
href	A required attribute, href refers to the URL link for the bookmark.
title	A required attribute, Title refers to the name of the link to be presented in the portal view.
target	A optional attribute, target refers to the name of a new window to create (or reuse) when opening the URL so that the current portal window is not replaced. The value of target should not contain spaces.

Displaying Bookmarks in a Portal View Through Direct Editing of XML

To add or change bookmarks that are displayed in a portal, you must edit the associated portal view file. Portal view files are located under the following directory:

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

When directly editing XML files, see "Rules for Direct Editing of XML Files" on page 312. If you need information on the XML elements, see "Bookmarks DTD" on page 351, "Shared Bookmarks DTD" on page 346, and "Portal View DTD" on page 142.

Here are three important terms you need to understand:

- `Entry`: A bookmark `Entry` is a single hyperlink that will be displayed in a portal view.

- `Group`: A bookmark `Group` is a logical grouping of entries. It consists of a `Group` element, which in turn contains a list of `Entry` elements described above. When displayed in a portal, this set of entries is preceded by the name of the group.

NOTE   A group can only be created through direct editing of the XML, not through the `Bookmarks` tab in the `SIP Administration Pages`.

- `SharedGroup`: A bookmark `SharedGroup` functions the same as the `Group` described above, with one distinction: instead of including the Entry list within the portal view file, a `SharedGroup` is a link to a bookmark group contained in a central location. This facilitates easy updates to many customer portals simultaneously. One may add the `SharedGroup` to a set of customers and then update the single copy to update the group for all of them.

Adding a Bookmark Entry to a Portal View

1. Using an ASCII editor, open the portal view file to which you want to add a bookmark entry.

2. Find the `Bookmarks` module and add the `Entry` element as shown in the following example:

```
<ModuleInstance
classid="com.hp.ov.portal.modules.bookmarks"
        display="yes" id="BookmarksModule-Demo1"
        rollupState="down">
    <Bookmarks>
        <Entry href="http://www.hp.com/e-services"
            target="bookmarkwin"
            title="Generic Net Primary Site"/>
        <Entry href="http://www.hp.com" target="bookmarkwin"
            title="HP Primary Site"/>
        <Entry href="http://www.openview.hp.com"
            target="bookmarkwin"
            title="HP OpenView Site"/>
    </Bookmarks>
```

The `Entry` element has three attributes:

- `href` - The URL link for the bookmark.

- `title` - The name of the link to be presented in the portal view.

- `target` - The name of a new window to create (or reuse) when opening the URL so that the current portal window is not replaced. Optional. The value of `target` should not contain spaces.

3. When you are finished, save and close the portal view file.

Grouping Entries

You can organize entries into a group under a group title. The `Group` element has one attribute, the `title`:

- `title` - Displayed before any contained `Entry` elements in an alternate style as a heading for the group.

See the example below:

```
<ModuleInstance classid="com.hp.ov.portal.modules.bookmarks"
        display="yes" id="BookmarksModule-Demo1"
        rollupState="down">
  <Bookmarks>
    <Group title="My Favorite OpenView Products">
      <Entry title="Service Information Portal"
       href="http://openview.hp.com/products/servinfoportal/index.asp"
```

```
        target="bookmarkwin"/>
   <Entry title="PolicyXpert"
     href="http://openview.hp.com/products/policyexpert/index.asp"
      target="bookmarkwin"/>
  <Entry title="Network Node Manager"
    href="http://openview.hp.com/products/nnm/index.asp"
    target="bookmarkwin"/>
  </Group>
</Bookmarks>
```

Changing the Order of Bookmarks Using XML

The portal displays bookmarks in the order in which they appear in the portal view file. To change the order, edit the portal view file (or, as appropriate, the `sharedBookmarks.xml`) to rearrange the order of the bookmark `Entry`, `Group`, and `SharedGroup` elements.

1. Using an ASCII editor, open the portal view file that has the Bookmarks module you want to modify.

2. Find the `Bookmarks` module and reorder the `Entry` elements, as desired.

3. When you are finished, save and close the portal view file.

4. To change the content of a `SharedGroup`, you need to modify it in the `sharedBookmarks.xml` file located in the following directory:

   Windows: `%SIP_HOME%\conf\share\modules\bookmarks\`
   UNIX: `/opt/OV/SIP/conf/share/modules/bookmarks/`

Removing Bookmarks From a Portal Using XML

To remove bookmarks from a portal view, edit the portal view file and delete any `Entry`, `Group`, or `SharedGroup` that you want to remove. If you think you might want to use the bookmark again sometime, instead of deleting it, comment out the element by preceding it with "<!--" and ending it with "-->".

1. Using an ASCII editor, open the portal view file that has the Bookmarks module you want to remove.

2. Find the `Bookmarks` module and delete the `Entry` elements, as desired.

3. When you are finished, save and close the portal view file.

4. To delete the content of a SharedGroup, you need to modify it in the sharedBookmarks.xml file located in the following directory:

Windows: %SIP_HOME%\conf\share\modules\bookmarks\
UNIX: /opt/OV/SIP/conf/share/modules/bookmarks/

NOTE   Any changes to this central file will affect all user roles that share this group. If you want to modify the group for one portal view only, you need to either copy the Group into the portal view file associated with the role or create a new SharedGroup for this role.

Bookmarks DTD

The OVBookmarks.dtd is duplicated below and stored in the following directory:

Windows: %SIP_HOME%\conf\share\views\
UNIX: /opt/OV/SIP/conf/share/views/

Following is a complete module instance that can be inserted directly into a portal view file. This code is automatically inserted into a portal view when a new bookmarks module is added from the portal interface.

Figure F-4   Default Bookmarks Module Instance

```
<Bookmarks>
   <Entry title="HP Primary Site" href="http://www.hp.com"
         target="bookmarkwin"/>
   <Entry title="HP OpenView Site" href="http://www.openview.hp.com"
         target="bookmarkwin"/>
   <Entry title="Service Information Portal"
         href="http://openview.hp.com/products/servinfoportal/index.asp"
         target="bookmarkwin"/>
   <SharedGroup name="hpsipintegrations"/>
</Bookmarks>
```

Table F-4   **Child Elements of the Bookmarks Element**

Element	Description
Entry	A single hyperlink that will be displayed in a portal view.
Group	A logical grouping of entries. Contains Entry elements that are grouped under a title. ("group" is the only bookmarks element that can only not be created through the Bookmarks tab in the SIP Administration Pages. It can only be created through direct editing of the XML.)
SharedGroup	Functions like the Group element above, with one distinction: instead of including the Entry list within the portal view file, a SharedGroup is a link to a bookmark group contained in a central location. This facilitates easy updates to many portals simultaneously. You may add the SharedGroup to multiple portal views and then update the single file to update the group for all of them.

Table F-5   **Entry Attributes**

Parameter	Description
title	The name of the link to be presented in the customer's portal.
href	The URL link for the bookmark.
target	Optional. Provides the name of a new window to create (or reuse) when opening the URL so that the current portal window is not replaced. The value of target should not contain spaces.

Table F-6   **Group Attributes**

Parameter	Description
title	Displayed before any contained `Entry` elements in an alternate style as a heading for the group.

Table F-7   **SharedGroup Attributes**

Parameter	Description
name	The name of a `SharedGroup` listed in the portal view file. When the portal finds the matching group in the `sharedBookmarks.xml` file, it interprets the matching group as a local bookmark group. `SharedGroup` can consist of zero or more `Group` definitions.

G   Uninstalling Service
         Information Portal

Uninstalling HP OpenView Service Information Portal

Be aware that uninstalling SIP will remove all of your portal configurations. If you want to save your customized files and directories, see *SIP Installation Guide* (SIP_Install_Guide.pdf) BEFORE uninstalling SIP.

NOTE   After uninstalling SIP, you may find that some installation logs remain on the file system. If you prefer to remove them, go to the SIP home directory and delete them.

On Windows

The state of your IIS Admin Service after uninstall will be the same as before uninstall.

1. Start: Settings->Control Panel->Add/Remove Programs.

2. Scroll down and select the version of SIP that you want to uninstall, and click [Add/Remove].

3. A prompt indicates that all files will be removed. Answer "yes" to continue uninstalling SIP.

On HP-UX and Solaris

1. As root, uninstall the software by running the command:
   /opt/OV/SIP/install/removesip

2. When prompted to continue with the removal, type "y" and press **Enter**.

Glossary

A-B

authentication The process by which a user identifies and validates him/herself to the system.

authentication provider A configured component of the system that authenticates users that attempt to use the system.

authorization The granting of access privileges to an authenticated user that determines what the user can see and do while logged into the system.

C

configuration The combination of settings of software parameters and attributes that determine the way the software works, the way it is used, and/or how it appears.

configuration file A file that contains specifications or information that can be used for determining how a software program should look and operate.

configure To define and/or modify specified software settings to fulfill the requirements of a specified environment, application and/or usage.

customer data filtering The use of attributes as a mask for constraining the data that is to be acted on, used, or displayed in the user interface. In HP OpenView Service Information Portal, the information that is presented in the user interface for a given user login is filtered by the specific management data that is associated with a given role. Customer data filtering can also be described as "customer segmentation."

customer model A mapping of customers to resources, where resources are associated hosts, interfaces, and services. A customer model can be defined in several XML files, or a mix of programs and files.

SIP 2.0 uses the so-called "Simple Customer Model" that is defined via an XML DTD.

customer model data source A configured URL or file that provides the mappings or partial mappings of customers to resources.

customize To design, construct and/or modify software to meet the needs and preferences of a particular customer or user. For HP OpenView Service Information Portal, customizing is synonymous with assigning to customers what will be displayed to them. Customization tasks include customizing content, tabs, and tab layout, customer filtering, and the setting of options.

customization The process of designing, constructing and/or modifying software to meet the needs and preferences of a particular customer or user.

E

edit permissions That which determines the editing operations that are available to a user through the program interface.

edit permissions level A group of operations that a user is authorized to perform through the program interface. Each level includes all the operations defined by the previous level and adds some additional operations.

extensible Software functionality whose capability, scope or effectiveness can be increased.

extend The act of increasing the capabilities, scope, and/or effectiveness of a program. The capabilities of HP OpenView Service Information Portal can be extended through the generic module and through the writing of XML.

F

filter A set of attributes and values that act as a pattern or mask through which data is passed. Filters allow matching-relevant information to be extracted and acted on while non-matching-irrelevant information is blocked.

M

Management Data Filter The security mechanism that defines what data is displayed through the portal.

message A communication using text and/or images. In HP OpenView Service Information Portal, messages are presented to customers via the message board module.

Message Board A module that is used for presenting messages to customers.

message content The information that is presented in a message. Message content may include text and/or graphics.

module A self-contained software component that performs a specific type of task or provides for the presentation of a specific type of data. Modules can interact with one another and with other software. In HP OpenView Service Information Portal,

modules present specific sets of functionality to the user through the portal framework. Examples of modules include the Message Board, Service Browser, Network Device Health, and the Alarm Module.

P

page A single display or presentation of information on the World Wide Web. Typically a web page consists of an HTML file, referenced graphics files, and associated scripts.

portal A web site that provides a variety of different types of information and which serves as a gateway to other web sites. HP OpenView Service Information Portal consists of the framework and modules. It provides information and access to other websites through the modules and submodules.

portal framework A program that acts as the basic structure to support other software modules or programs that provide additional functionality for the user. In HP OpenView Service Information Portal, the framework provides a mechanism for the modules to present information to the user. The framework also provides the structure for customization, configuration and extension of the portal's functionality.

portal view Consists of modules, tabs, and view properties. Each user account that is set up by an administrator has one or more roles, and each role is associated with one portal view. A portal view can be shared by multiple users.

portal view file A configuration file that contains specifications or information that can be used for determining how software should look to a given user. In HP OpenView

Service Information Portal, portal view files are XML files that contain all information needed to render a portal view.

R

role That which defines what a user can see and do through the portal at a particular point in time. A role can be shared by multiple users.

role properties An extensibility mechanism used to provide authorization information associated with a role and that is not defined in the predefined role XML elements.

S

skin A setting that controls the visual appearance of the user interface. Skins can determine the color scheme, fonts, graphics, and other attributes presented in the user interface. The skins in the HP OpenView Service Information Portal are based upon W3C's Cascading Style Sheets. Existing skins may be extended, or new ones added to the 'css' files located under the `htdocs/styles` directory.

submodule A portion of a software module that provides a subset of the functionality provided by the module. A submodule performs a specific task or presents a specific set of data. In HP OpenView Service Information Portal submodules present different variations of the type of data presented by the Module. For example, one submodule of the Network Device Health Module presents Network health for Routers while another submodule presents Network health for Servers.

T

tab A page in the user interface that has a small index-card like projection. The projection typically presents the name for the page and allows navigation to the page by clicking. In HP OpenView Service Information Portal the main portal pages have tabs. Service Information Portal provides one tab, but multiple tabs can be created using the Customize Content option on the Options page. Similar types of modules can be grouped together using tabs.

U

user preferences The attributes that are associated with a specific user. In Service Information Portal, user preferences control the name that appears in the portal header, and the color scheme, or "skins" that control the portal colors and fonts.

User-Role Model An authorization model that achieves security by associating users with roles and assigning to each role what the user is able to see and do. The User-Role Model consists of all User Role Package files.

User-Role Model

Index

Index

Index

Index

Index

Index