

HP OpenView Operations

Service Navigator

Concepts and Configuration Guide

Software Version: A.08.10

UNIX



Manufacturing Part Number: B7491-90048

October 2004

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

Legal Notices

Warranty.

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

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

Restricted Rights Legend.

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Hewlett-Packard Company
United States of America

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

Copyright Notices.

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

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

Trademark Notices.

Adobe® is a trademark of Adobe Systems Incorporated.

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

Intel386, Intel80386, Intel486, and Intel80486 are U.S. trademarks of Intel Corporation.

Intel Itanium™ Logo: Intel, Intel Inside and Itanium are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries and are used under license.

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

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

MS-DOS® is a U.S. registered trademark of Microsoft Corporation.

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

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

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

OSF, OSF/1, OSF/Motif, Motif, and Open Software Foundation are trademarks of the Open Software Foundation in the U.S. and other countries.

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

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

UNIX® is a registered trademark of the Open Group.

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

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

1. HP OpenView Service Navigator

| | |
|--|----|
| In This Chapter | 20 |
| What is HP OpenView Service Navigator? | 21 |
| What Does Service Navigator Do? | 23 |
| Impacted Services Analysis | 23 |
| Root Cause Analysis | 24 |
| How Does Service Navigator Work? | 26 |
| What is a Service Hierarchy? | 26 |
| What is the Status of a Service? | 29 |
| Status Propagation Rules | 31 |
| Status Calculation Rules | 34 |
| Weighting | 36 |
| How Does OVO Map Messages to Services? | 42 |
| The Service Navigator GUI | 44 |
| Service Graphs | 45 |
| Service Submaps | 46 |
| Enhancing Service Submaps | 47 |
| Custom Service Maps | 48 |
| Service Properties | 49 |
| Labeling Services | 51 |
| Loading Services on Demand | 53 |
| Service Logging and Reporting | 54 |

2. Getting Started

| | |
|--|----|
| In This Chapter | 58 |
| How Do I Install Service Navigator? | 59 |
| Installing Service Navigator | 59 |
| Determining the size of max_thread_proc | 60 |
| How Do I Configure Service Navigator? | 61 |
| Planning Your Service Hierarchy | 62 |
| Writing the Service Configuration File | 64 |
| Activating the Service Configuration | 66 |
| Modifying the Service Configuration | 66 |
| Service Files Error Checking | 67 |
| Setting the Service Name in OVO | 67 |
| Assigning Services to Operators | 71 |
| Assigning Services with opcservice | 72 |
| Assigning Services in the Configuration File | 72 |

| | |
|---|----|
| Planning Your Service Hours | 73 |
| Enabling Service Logging | 74 |
| Setting Service Attributes Dynamically | 75 |
| What to be Aware of When Setting Service Attributes Dynamically | 76 |
| Setting Service Attributes Dynamically with opcsvcatr | 77 |
| Setting Service Attributes Dynamically with OVO Messages | 79 |
| How Do I Start Service Navigator? | 93 |
| How Do I Upgrade Service Navigator? | 94 |
| How Do I De-install Service Navigator? | 96 |
| To De-install the Service Navigator GUI Client | 96 |
| To De-install Service Navigator from an OVO Management Server | 97 |

3. The Service Configuration File

| | |
|---|-----|
| In This Chapter | 100 |
| Creating the Service Hierarchy | 101 |
| Defining the Rules | 104 |
| Setting up Service Actions | 107 |
| Using Attributes in Service Actions | 108 |
| Setting up Local Actions | 108 |
| Setting Service Attributes | 109 |
| Setting up Service Assignments | 111 |

A. Did You Know...

| | |
|--|-----|
| In This Chapter | 114 |
| ... That Names are Important? | 115 |
| ... How to Use OVO Variables? | 118 |
| ... That You Can Use Example Configuration Files? | 119 |
| ... That You Can Customize Icons and Backgrounds? | 122 |
| ... How to Label Service Icons? | 124 |
| Syntax for Dynamic Labels | 125 |
| Text Labels | 125 |
| Image Labels | 125 |
| An Example for Setting Labels Dynamically | 126 |
| ... How to Enhance a Service Submap? | 128 |
| ... That You Can Operate with Service Navigator GUI From Other Applications? | 130 |
| ... That You Can Personalize Your Service Graph and Service Map View? | 131 |

| | |
|--|-----|
| Service Graph and Map Zooming | 131 |
| Service Graph and Map Quick Zooming | 132 |
| Setting Default Zoom Ratio | 133 |
| Navigating through Service Graphs and Maps | 134 |
| Navigating Using Navigation Panel | 134 |
| Navigating from the Toolbar | 135 |
| Making Selections in Service Graphs | 136 |
| Easy Scrolling and Repositioning of Services in Service Graphs | 138 |
| Customizing Font Size in Service Graphs and Maps | 140 |

B. Reference Information

| | |
|---|-----|
| In This Chapter | 142 |
| The Service Configuration File Syntax | 143 |
| Notation Used | 144 |
| Defining the Services and Associations | 151 |
| Defining Status Calculation and Propagation Rules | 158 |
| Defining Service Actions | 162 |
| Assigning Services to Operators | 164 |
| Reserved Service Attributes | 165 |
| The opcservice Command | 168 |
| OVO Variables | 173 |
| OVO Variables in Service Names | 173 |
| OVO Variables in Service Actions | 173 |
| OVO Variables in URL Definitions | 175 |
| Service Navigator Man Pages | 176 |
| Service Navigator Product Bundles | 177 |

| | |
|---------------------------|------------|
| Glossary | 179 |
|---------------------------|------------|

| | |
|------------------------|------------|
| Index | 181 |
|------------------------|------------|

Printing History

The manual printing date and part number indicate its current edition. The printing date will change when a new edition is printed. Minor changes may be made at reprint without changing the printing date. The manual part number will change when extensive changes are made.

Manual updates may be issued between editions to correct errors or document product changes. To ensure that you receive the updated or new editions, you should subscribe to the product support service. See your HP sales representative for details.

| | |
|------------------|----------------|
| First Edition: | February 1999 |
| Second Edition: | September 1999 |
| Third Edition: | June 2000 |
| Fourth Edition: | January 2002 |
| Fifth Edition: | April 2002 |
| Sixth Edition: | August 2002 |
| Seventh Edition: | June 2004 |
| Eighth Edition: | October 2004 |

Conventions

The following typographical conventions are used in this manual.

Table 1 **Typographical Conventions**

| Font | Meaning | Example |
|---------------|--|--|
| <i>Italic</i> | Book or manual titles, and man page names | Refer to the <i>OVO Administrator's Reference</i> and the <i>opc(1M)</i> manpage for more information. |
| | Emphasis | You <i>must</i> follow these steps. |
| | Variable that you must supply when entering a command | At the prompt, enter rlogin <i>username</i> . |
| | Parameters to a function | The <i>oper_name</i> parameter returns an integer response. |
| Bold | New terms | The HTTPS agent observes... |
| Computer | Text and other items on the computer screen | The following system message displays: Are you sure you want to remove current group? |
| | Command names | Use the <code>grep</code> command ... |
| | Function names | Use the <code>opc_connect()</code> function to connect ... |
| | File and directory names | <code>/opt/OV/bin/OpC/</code> |
| | Process names | Check to see if <code>opcmona</code> is running. |
| | Window/dialog-box names | In the Add Logfile window ... |
| | Menu name followed by a colon (:) means that you select the menu, then the item. When the item is followed by an arrow (->), a cascading menu follows. | Select Actions: Filtering -> All Active Messages from the menu bar. |

Table 1 **Typographical Conventions (Continued)**

| Font | Meaning | Example |
|--------------------------|-------------------------------|-----------------------------------|
| Computer Bold | Text that you enter | At the prompt, enter ls -l |
| Keycap | Keyboard keys | Press Return . |
| [Button] | Buttons in the user interface | Click [OK]. |

OVO Documentation Map

HP OpenView Operations (OVO) provides a set of manuals and online help that help you to use the product and to understand the concepts underlying the product. This section describes what information is available and where you can find it.

Electronic Versions of the Manuals

All the manuals are available as Adobe Portable Document Format (PDF) files in the documentation directory on the OVO product CD-ROM.

With the exception of the *OVO Software Release Notes*, all the manuals are also available in the following OVO web-server directory:

`http://<management_server>:3443/ITO_DOC/<lang>/manuals/*.pdf`

In this URL, `<management_server>` is the fully-qualified hostname of your management server, and `<lang>` stands for your system language, for example, `C` for the English environment and `japanese` for the Japanese environment.

Alternatively, you can download the manuals from the following website:

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

Watch this website regularly for the latest edition of the OVO Software Release Notes, which gets updated every 2-3 months with the latest news such as additionally supported OS versions, latest patches and so on.

OVO Manuals

This section provides an overview of the OVO manuals and their contents.

Table 2 **OVO Manuals**

| Manual | Description | Media |
|---|--|-----------------|
| <i>OVO Installation Guide for the Management Server</i> | <p>Designed for administrators who install OVO software on the management server and perform the initial configuration.</p> <p>This manual describes:</p> <ul style="list-style-type: none">• Software and hardware requirements• Software installation and de-installation instructions• Configuration defaults | Hardcopy PDF |
| <i>OVO Concepts Guide</i> | <p>Provides you with an understanding of OVO on two levels. As an operator, you learn about the basic structure of OVO. As an administrator, you gain an insight into the setup and configuration of OVO in your own environment.</p> | Hardcopy PDF |
| <i>OVO Administrator's Reference</i> | <p>Designed for administrators who install OVO on the managed nodes and are responsible for OVO administration and troubleshooting. Contains conceptual and general information about the OVO DCE/NCS-based managed nodes.</p> | PDF only |
| <i>OVO DCE Agent Concepts and Configuration Guide</i> | <p>Provides platform-specific information about each DCE/NCS-based managed-node platform.</p> | PDF only |
| <i>OVO HTTPS Agent Concepts and Configuration Guide</i> | <p>Provides platform-specific information about each HTTPS-based managed-node platform.</p> | PDF only |
| <i>OVO Reporting and Database Schema</i> | <p>Provides a detailed description of the OVO database tables, as well as examples for generating reports from the OVO database.</p> | PDF only |
| <i>OVO Entity Relationship Diagrams</i> | <p>Provides you with an overview of the relationships between the tables and the OVO database.</p> | PDF only |

Table 2 **OVO Manuals (Continued)**

| Manual | Description | Media |
|--|--|-----------------|
| <i>OVO Java GUI Operator's Guide</i> | Provides you with a detailed description of the OVO Java-based operator GUI and the Service Navigator. This manual contains detailed information about general OVO and Service Navigator concepts and tasks for OVO operators, as well as reference and troubleshooting information. | PDF only |
| <i>Service Navigator Concepts and Configuration Guide</i> | Provides information for administrators who are responsible for installing, configuring, maintaining, and troubleshooting the HP OpenView Service Navigator. This manual also contains a high-level overview of the concepts behind service management. | Hardcopy PDF |
| <i>OVO Software Release Notes</i> | Describes new features and helps you: <ul style="list-style-type: none">• Compare features of the current software with features of previous versions.• Determine system and software compatibility.• Solve known problems. | PDF only |
| <i>OVO Supplementary Guide to MPE/iX Templates</i> | Describes the message source templates that are available for the MPE/iX managed nodes. This guide is not available for OVO on Solaris. | PDF only |
| <i>Managing Your Network with HP OpenView Network Node Manager</i> | Designed for administrators and operators. This manual describes the basic functionality of the HP OpenView Network Node Manager, which is an embedded part of OVO. | Hardcopy PDF |
| <i>OVO Database Tuning</i> | This ASCII file is located on the OVO management server at the following location: /opt/OV/ReleaseNotes/opc_db.tuning | ASCII |

Additional OVO-related Products

This section provides an overview of the OVO-related manuals and their contents.

Table 3 **Additional OVO-related Manuals**

| Manual | Description | Media |
|--|--|-----------------|
| HP OpenView Operations for UNIX Developer's Toolkit If you purchase the HP OpenView Operations for UNIX Developer's Toolkit, you receive the full OVO documentation set, as well as the following manuals: | | |
| <i>OVO Application Integration Guide</i> | Suggests several ways in which external applications can be integrated into OVO. | Hardcopy PDF |
| <i>OVO Developer's Reference</i> | Provides an overview of all the available application programming interfaces (APIs). | Hardcopy PDF |
| HP OpenView Event Correlation Designer for NNM and OVO If you purchase HP OpenView Event Correlation Designer for NNM and OVO, you receive the following additional documentation. Note that HP OpenView Event Correlation Composer is an integral part of NNM and OVO. OV Composer usage in the OVO context is described in the OS-SPI documentation. | | |
| <i>HP OpenView ECS Configuring Circuits for NNM and OVO</i> | Explains how to use the ECS Designer product in the NNM and OVO environments. | Hardcopy PDF |

OVO Online Information

The following information is available online.

Table 4 **OVO Online Information**

| Online Information | Description |
|--|--|
| HP OpenView Operations Administrator's Guide to Online Information | Context-sensitive help system contains detailed help for each window of the OVO administrator Motif GUI, as well as step-by-step instructions for performing administrative tasks. |
| HP OpenView Operations Operator's Guide to Online Information | Context-sensitive help system contains detailed help for each window of the OVO operator Motif GUI, as well as step-by-step instructions for operator tasks. |
| HP OpenView Operations Java GUI Online Information | HTML-based help system for the OVO Java-based operator GUI and Service Navigator. This help system contains detailed information about general OVO and Service Navigator concepts and tasks for OVO operators, as well as reference and troubleshooting information. |
| HP OpenView Operations Man Pages | <p>Manual pages available online for OVO. These manual pages are also available in HTML format.</p> <p>To access these pages, go to the following location (URL) with your web browser:</p> <p><code>http://<management_server>:3443/ITO_MAN</code></p> <p>In this URL, the variable <code><management_server></code> is the fully-qualified hostname of your management server. Note that the man pages for the OVO HTTPS-agent are installed on each managed node.</p> |

1 HP OpenView Service Navigator

In This Chapter

This section describes what Service Navigator is and how it works.

NOTE

It is essential for successful deployment and operation of this product that you understand its underlying concepts. This applies in particular to the status propagation and calculation rules.

This guide assumes that you are a knowledgeable system and network administrator, and that you have previous experience with HP OpenView HP OpenView Operations.

What is HP OpenView Service Navigator?

HP OpenView Service Navigator is an add-on component of the HP OpenView HP OpenView Operations Java-based operator GUI. It enables you to manage your IT environment while focusing on the IT services you provide.

While HP OpenView Operations lets you detect, solve, and prevent problems occurring in networks, systems and applications in your IT environment, Service Navigator takes you one step further. Service Navigator lets you map the problems discovered by OVO to the IT services you want to monitor. Instead of focusing on single elements within a complex IT environment, you can now manage your IT environment by focusing on the IT services you are responsible for.

Service Navigator is based on OVO and depends on the monitoring, message, and action capabilities OVO provides. If a problem occurs on one of the objects managed by OVO, a message about this problem is generated and sent to the user responsible for the area concerned. With Service Navigator installed and configured, this message is mapped to the service that is affected by the problem, and sent to the user responsible for that service.

The severity status of the problem also changes the severity status of the service so that the user can easily identify services that are in a problematic state. To solve service-related problems, OVO's problem resolution capabilities have been further extended to include service-specific analysis operations and actions.

If enabled, Service Navigator logs each change of status in the database so that reports about service availability can be generated.

Figure 1-1 on page 22 shows the main window of Service Navigator. In addition to the customary OVO managed nodes and message groups, managed services are displayed in the scoping pane on the left. The content area on the right is split into two sections. The upper section displays a service hierarchy in graphical form where each service is represented by an icon. The lower section contains the standard OVO message browser configured to display only the messages that pertain to your service.

HP OpenView Service Navigator
 What is HP OpenView Service Navigator?

Figure 1-1 The Service Navigator GUI

The screenshot displays the HP OpenView Service Navigator interface. At the top, a menu bar includes 'File', 'Edit', 'View', 'Actions', 'Window', and 'Help'. Below the menu is a toolbar with various icons. The main interface is divided into several panes:

- Left Pane (Shortcuts to Services):** A vertical list of service categories including Banking Services, E-Mail, SAP R/3, and Online Help.
- Center Pane (OVO Managed Nodes):** A hierarchical tree view showing the structure of managed services. It includes categories like Application Servers, DB Systems, External, Message Groups, Applications, Services, and SAP R/3. Under 'Application Servers', it lists 'Application Instance inst1' and its sub-components: Operation System, Abap, Buffers, Display Queues, Performance, RollPaging, Processes, and Tablespace.
- Right Pane (Service Graph):** A graphical representation of the service dependencies. It shows 'Application Server' at the top, connected to 'Application Instance inst1', which in turn connects to various sub-components like 'Operation System', 'Abap', 'Buffers', 'Display Queues', 'Performance', 'RollPaging', 'Processes', and 'Tablespace'.
- Bottom Pane (Message Browser with Service Messages):** A table displaying a list of messages. The table has columns for Severity, DU, SUTAONE, Time Received, Node, Application, MsgGrp, Object, Message Text, Switch From, and Switch To. The messages include various alerts such as 'CPU Bottleneck probability' and 'Node down'.

At the bottom of the interface, there is a status bar showing '100 of 6 9 777 428 131 0 0 0' and a 'Ready' indicator. A label 'Managed Services' points to the central tree view.

What Does Service Navigator Do?

Service Navigator helps you to quickly identify and solve the most critical problems affecting your services before service performance deteriorates.

There are two ways of handling problems with Service Navigator:

- ❑ The traditional method of watching messages arrive in the message browser, then highlighting the service that is affected and performing a get impacted services analysis to find out which other services are affected.
- ❑ Following a service-oriented top-down approach: open a service graph and monitor the status of the displayed services; when a service changes status, perform a root cause analysis to find out where the cause of the problem lies.

Impacted Services Analysis

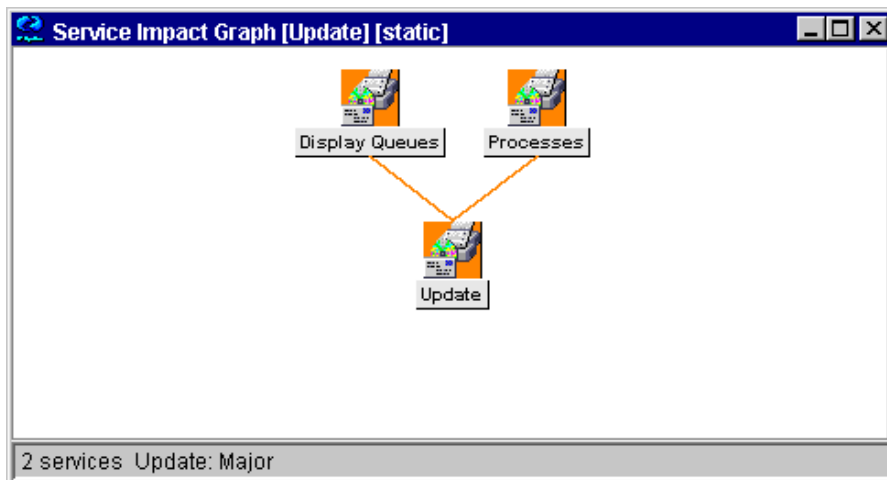
Whereas the root cause analysis drills down the service hierarchy to the service that caused the status to change, get impacted services works the other way: it searches upwards through the service hierarchy and displays all other services that are affected by the change in status. The nature of the effect is determined by considering the configured status propagation and calculation rules. This helps operators prioritizing their tasks. They can determine which problem has the highest impact, for example, a problem that impacts critical business services and a large number of users should be solved first. Get impacted services analysis identifies whether these critical business services are affected.

In Figure 1-2 on page 24, the subservice `Update` has the severity `major` which impacts the services `Display Queues` and `Processes`. Both services are affected by a problem related to `Update`. Note that the service impact graph, just like the root cause graph, is static; it does not get updated but displays the status of the services at the time it was generated.

Impact analysis is also available for messages: select a message in a browser and click `Get Impacted Services`. This highlights the affected services in the service tree in the scoping pane.

Note that impact analysis only considers negative effects, that is a status change from “good” to “bad”; for example, a status change from major to critical, but not from critical to major. This means that the impact graph does not necessarily display the toplevel service, but stops where the last service was impacted.

Figure 1-2 **Get Impacted Services Analysis**



Root Cause Analysis

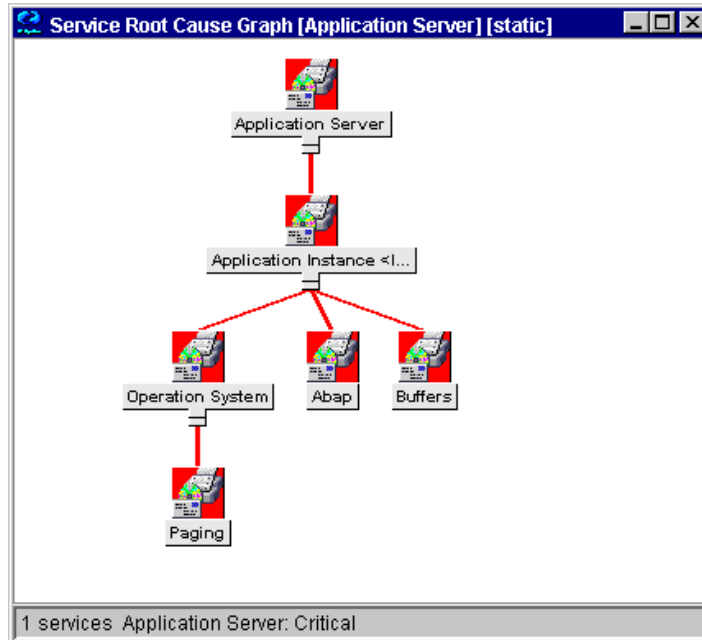
In a service hierarchy with many hierarchical levels it can be difficult to determine why a service has adopted a certain severity status. Did it turn critical because of a critical message, a critical subservice, or several affected subservices?

Service Navigator helps you answer these questions by performing a root cause analysis. You simply highlight the icon of the service you are interested in and select or click *Get Root Cause*. This opens a new window which displays the service you had selected and all other subservices that contributed to its status.

Root cause analysis performs a top-down investigation. It stops on the hierarchical level where the cause of the problem lies. Note that there may be more branches or levels below that which are not displayed because the services on these levels did not contribute to the problem.

See Figure 1-3 on page 25 for an example of a root cause graph. Note that the root cause graph is static; it does not get updated but displays the status of the services at the time it was generated.

Figure 1-3 Get Root Cause Analysis



How Does Service Navigator Work?

The concept behind Service Navigator is that of a hierarchical service structure. Service Navigator lets you build a hierarchy that reflects the relationships and dependencies between the service-relevant managed objects in your IT environment.

The following section explains the concept of a service hierarchy and how a service can be logically related to a subservice. “What is the Status of a Service?” on page 29 explains the dynamic aspects of the relationship between a service and its subservices in terms of status severity.

What is a Service Hierarchy?

A **service hierarchy** is a logical organization of the services you provide, each higher level covering a wider or more general service area than the next lower level.

The relationship between services in a hierarchy can be one of two types:

containment

A service is contained in a service; that is, a service is part of, and defined within, another service. The contained service cannot exist without the containing service. A service can contain more than one subservices.

usage

A service is contained in a service but, at the same time, is used, or referenced, by another service. The used service can exist without the using service; the using service depends on the used service.

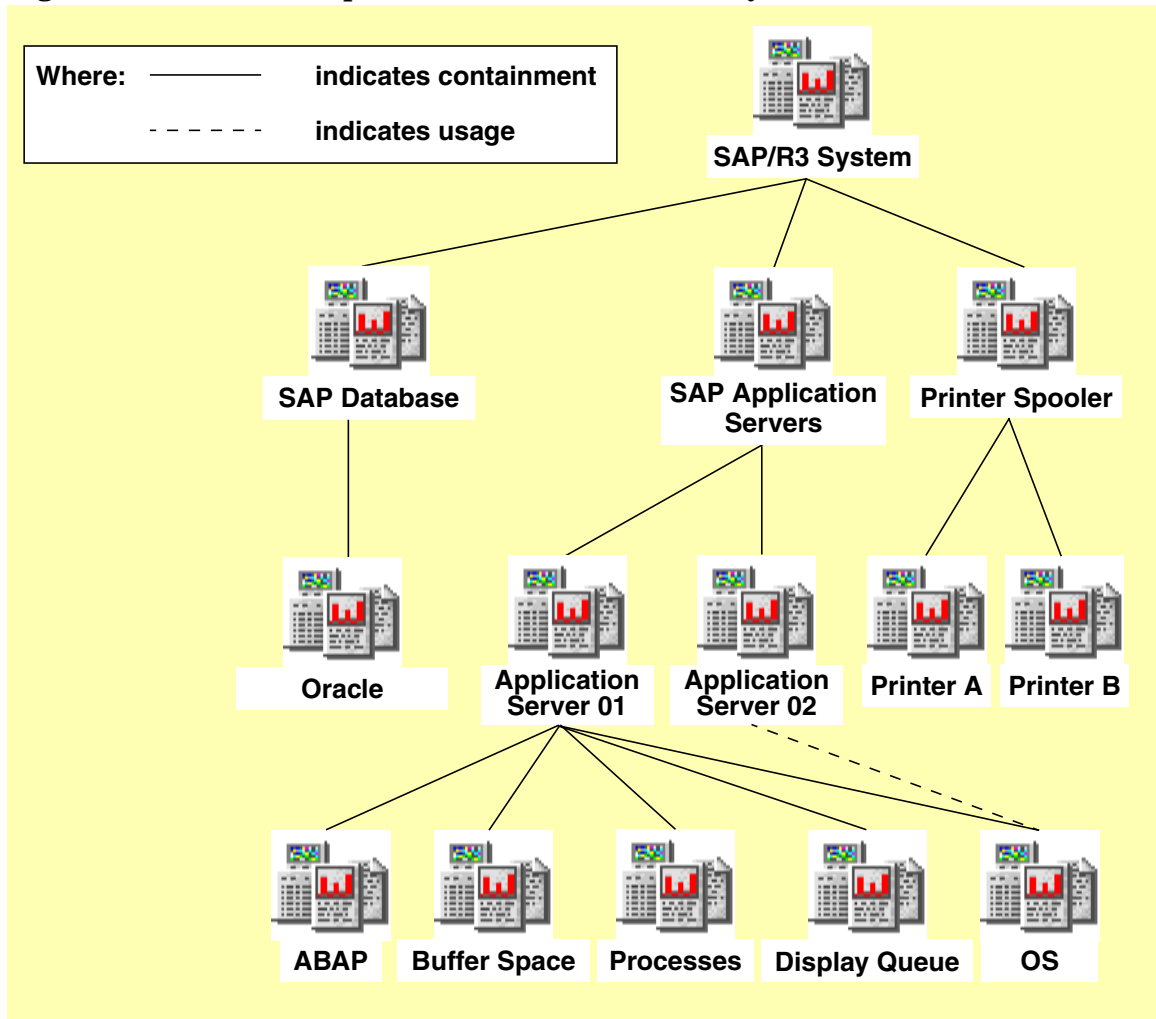
For the purposes of status propagation and calculation it is of no consequence whether a service is contained in or used by another service. Note, however, that a service can be defined only once but be used or contained many times.

Service Navigator supports up to 256 hierarchical levels.

Figure 1-4 on page 28 shows an example of a service hierarchy for an SAP/R3 system. The SAP/R3 service includes components of SAP database servers and application servers, as well as a printer spooler. Each of these subservices is divided into further elements. Together they build the SAP/R3 service hierarchy. The relationship between the services is of the type *containment*, except for the relationship between the Application Server 02 and its subservice. Application Server 02 *uses* a subservice contained in Application Server 01.

The fact that OVO allows one service using another subservice, saves you having to set up specific subservices for each of your service hierarchies. You can set up a generic service, for example monitoring the operating system on an SAP/R3 application server, that can be used by any other service hierarchy responsible for monitoring an SAP/R3 application server.

Figure 1-4 Example: SAP/R3 Service Hierarchy

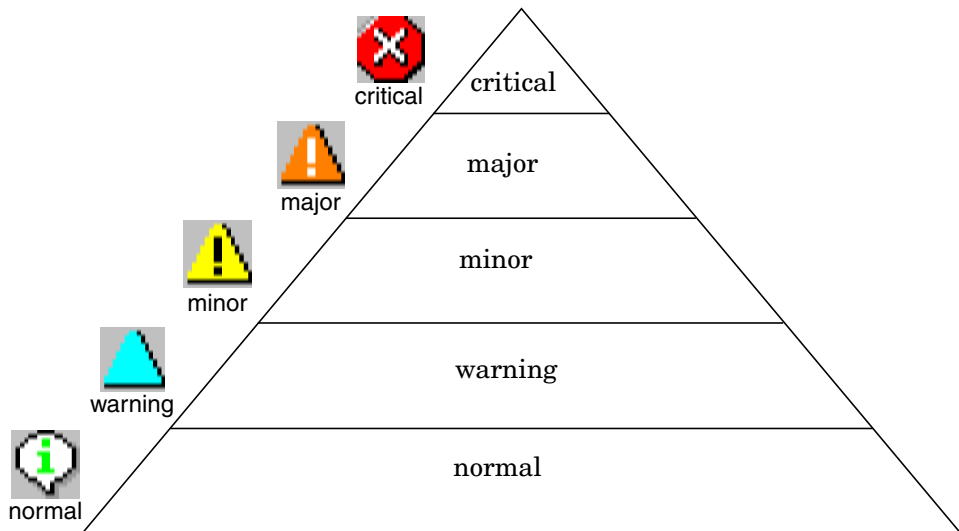


What is the Status of a Service?

The status of a service is the current operational status of a service. The status is indicated by status colors, each color signalling how severe the current situation is. For example, the color red indicates a critical problem situation. See also Figure 1-5 for an illustration of the OVO severity status model.

Figure 1-5

OVO Severity Pyramid



Since a service contains or uses several subservices, and since each subservice in turn contains or uses other subservices, the severity status of the service must be determined and calculated on the basis of the severity statuses of the next lower services in addition to the severity statuses of the messages targeting the service.

This is done on the basis of status propagation and status calculation rules. These rules define the relationship of a service to its subservices in terms of how they interpret each other's severity status. Status rules are defined in the service configuration file; see "The Service Configuration File Syntax" on page 143 for more information.

An example: a service contains two subservices, one has the severity status critical, the other has the severity status normal. The service interprets these for what they are, and sets its own severity status to critical because a critical state is more serious than a normal state.

However, Service Navigator gives you much greater flexibility in terms of how the service interprets other severity statuses. For example, a subservice may be less important than another subservice; that is, problems in that subservice have a smaller effect on the service than problems in another subservice. In this case, you would want to decrease the severity status of the one subservice but increase the severity status of the other subservice. This is defined by applying different **status propagation rules** to the relationship between a service and a subservice; see also “Status Propagation Rules” on page 31.

Status calculation rules, on the other hand, define how a service computes its severity status from the severity statuses of its subservices after the status propagation rules have been applied. In the above example, the service sets its severity status to critical because one of its subservices is critical. This is an application of the status calculation rule **Most Critical**. You can, however, define other rules, as explained in “Status Calculation Rules” on page 34.

You can also make one service count more than another service by adding a **weighting factor** for the purposes of status calculation. See the section on “Weighting” on page 36 for more information.

Status Propagation Rules

Status propagation rules define how a service determines the severity of a subservice. The relationship between two services can be one of the following:

increase If the subservice is interpreted as more important, that is, its severity status has more effect on the next higher service than other subservices on the same hierarchical level, its severity status is increased by a level.

For example, if a subservice has the severity status minor, and status propagation is defined as increase, its severity status is increased from minor to major.

unchanged The subservice propagates its severity status unchanged to the service. This is the default configuration.

For example, if a subservice has the severity status normal, the service interprets it as normal.

decrease If the subservice is interpreted as less important, that is, its severity status has less effect on the next higher service than other subservices on the same hierarchical level, its severity status is decreased by a level.

For example, if a subservice has the severity status critical, and status propagation is defined as decrease, its severity status is decreased from critical to major.

ignore The severity of the subservice does not influence the service. Propagation does not take place.

setto The severity of the subservice is set to a specified severity. The new severity status can be either higher or lower than the original status.

For example, the severity of a subservice can always be set to the severity status critical.

See Figure 1-6 on page 33 for an illustration of how the selected status propagation rule influences the status of a service. In this example the subservices ABAP and Buffer Space are interpreted as more important than the subservices Processes and Display Queue. Therefore, the severity statuses of the former are increased, the severity statuses of the latter are decreased. The service Application Server now has a manipulated but correct view of the actual situation.

Status propagation rules can be set in two ways:

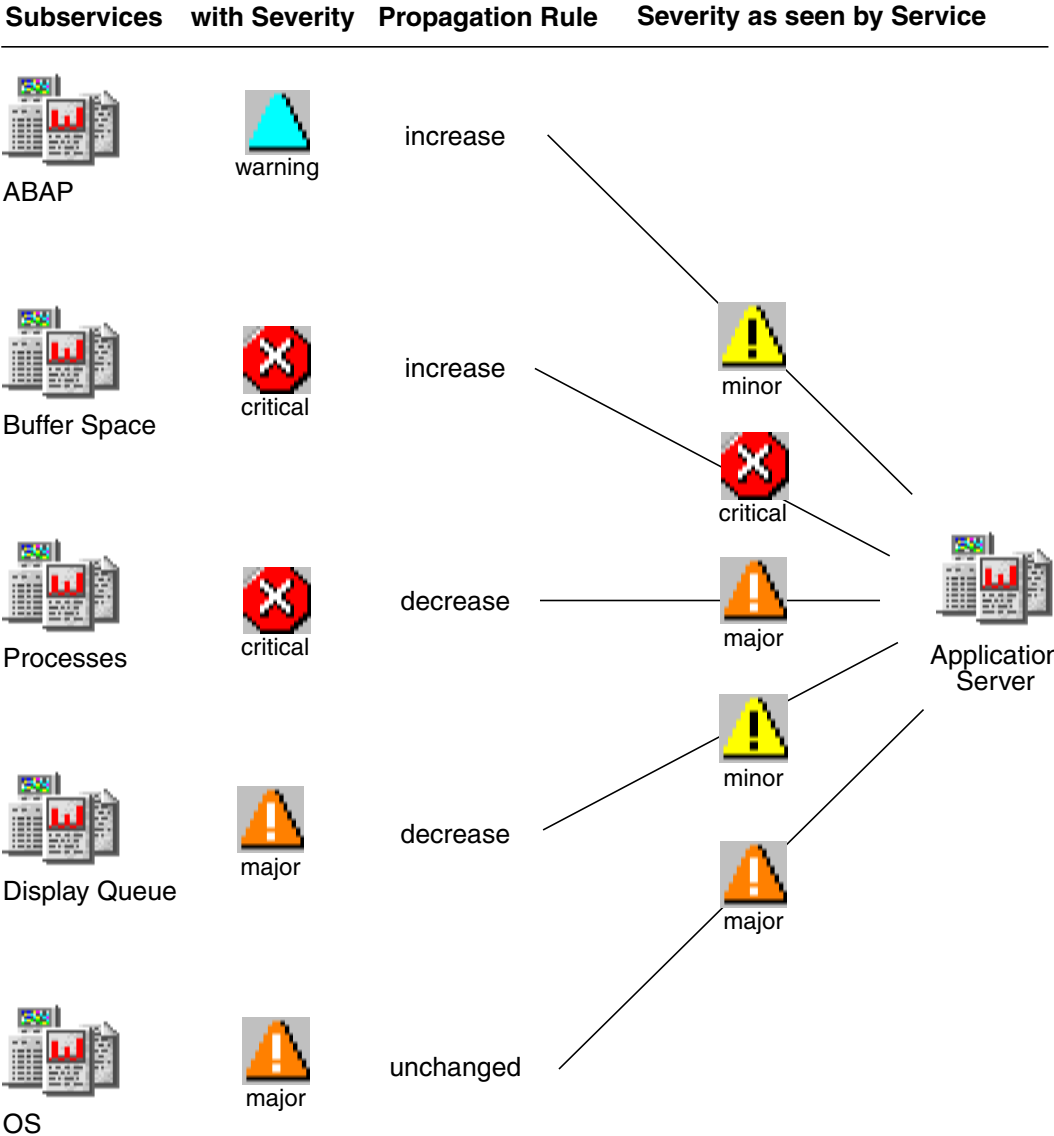
❑ **Default Propagation Attribute**

A default propagation attribute is applied; for example, the severity of a subservice is always increased.

❑ **Severity-specific Propagation Attribute**

Severity-specific propagation attributes are applied. This means that, depending on the severity of the subservice, a specific propagation rule is applied; for example, the severity normal is always ignored, but the severity warning is always increased.

Figure 1-6 Example: Status Propagation Rules



Status Calculation Rules

Status calculation rules define the formula for calculating the severity of a service based on the severity of its subservices and the messages targeting the service directly. Calculation rules model the behavior of the relationship between services. For example, by selecting a calculation rule, you determine, whether the highest severity of the subservices or the highest average severity of the subservices determines the severity of the service.

The concept is based on setting thresholds that must be exceeded before the status of the service changes. The service adopts the status of the subservice or message that first exceeds the threshold.

The following status calculation rules are available:

Most Critical The service adopts the status of the subservice with the highest severity. The absolute threshold is set to 0. (This is a special case of the Single Threshold rule.)

The Most Critical rule models a worst-case scenario: the highest severity of the subservices “wins”.

Single Threshold The highest severity of the subservices that is found more often than the threshold will be the severity of the service. (This is a special case of the Multiple Threshold rule where the same threshold is defined for all severities.)

The Single Threshold rule computes the highest average severity of all subservices. This average severity is then adopted by the service.

Multiple Thresholds The concept is the same as Single Threshold but you can set individual thresholds for each severity except the severity normal.

NOTE

It is *not* possible to specify more than one threshold for the same severity. If multiple thresholds are specified, only the last one will be active.






The Multiple Thresholds rule, for example, allows a few severe problems to be more important than many problems of a lower severity.

The threshold itself can be defined as an absolute value or as a percentage. See Appendix , “The Service Configuration File Syntax,” on page 143 for information on the format of a percentage that should be used when specifying a relative threshold value in the service configuration file.

When calculating the value that is compared to the threshold, the severity of the subservice and their lower severities are counted. For example, if a subservice has the severity critical, the severities major, minor, warning, and normal are also counted. See also Figure 1-5 on page 29.

Table 1-1 details how the severity of each subservice in Figure 1-6 and each message is calculated. The first column lists all subservices and messages. The second column states the severity of each subservice as it is interpreted by the service. This is determined by the status propagation rules explained in the previous section. The remaining columns are then compared to the severity of the subservice, and a 1 is entered if they match, a 0 if they do not match. You must also enter a 1 for each lower severity of a matching subservice.

Table 1-1 Example: Status Calculation Scenario

| Subservices | Severity as interpreted by the service Application Server |  critical |  major |  minor |  warning |  normal |
|--|---|---|--|--|--|---|
| ABAP | minor | 0 | 0 | 1 | 1 | 1 |
| Buffer Space | critical | 1 | 1 | 1 | 1 | 1 |
| Processes | major | 0 | 1 | 1 | 1 | 1 |
| Display Queue | minor | 0 | 0 | 1 | 1 | 1 |
| OS | major | 0 | 1 | 1 | 1 | 1 |
| Messages targeting Application Server: severity minor ^a | | 0 | 0 | 1 | 1 | 1 |
| Total: | | 1 (16%) | 3 (50%) | 6 (100%) | 6 (100%) | 6 (100%) |

- a. The messages directly targeting a service are seen as an entity: the message with the highest severity effects the service.

For example, the service Application Server interprets the severity of the subservice ABAP as minor. The original severity of the ABAP is warning but the relationship is defined as increase. Consequently the status normal is increased to minor (see also Figure 1-6). You enter a **1** in the column where minor matches minor, and a **1** for every subsequent lower severity, in this case for warning and normal.

When you have done this for all subservices, you add up each column to arrive at a total for each severity.

Depending on the status calculation rule and the threshold you have selected, the service Application Server changes severity. For example, if one of the following were configured:

- Most Critical: the service changes to critical because the highest severity found amongst the subservices and messages (critical) is adopted by the service.
- Single Threshold 50%: the service changes to major because the severity that first exceeds the threshold of 50% is adopted by the service. In this case the severity major with 50% is the first to exceed the threshold of 50%.
- Multiple thresholds:
 - threshold of 15% for severity critical
 - threshold of 50% for severity major
 - threshold of 60% for severity minor
 - threshold of 70% for severity warning

The service changes to critical because the first threshold to be exceeded is 15% (severity critical).

Note that there is no need to specify a threshold for the severity normal because normal is the lowest severity status.

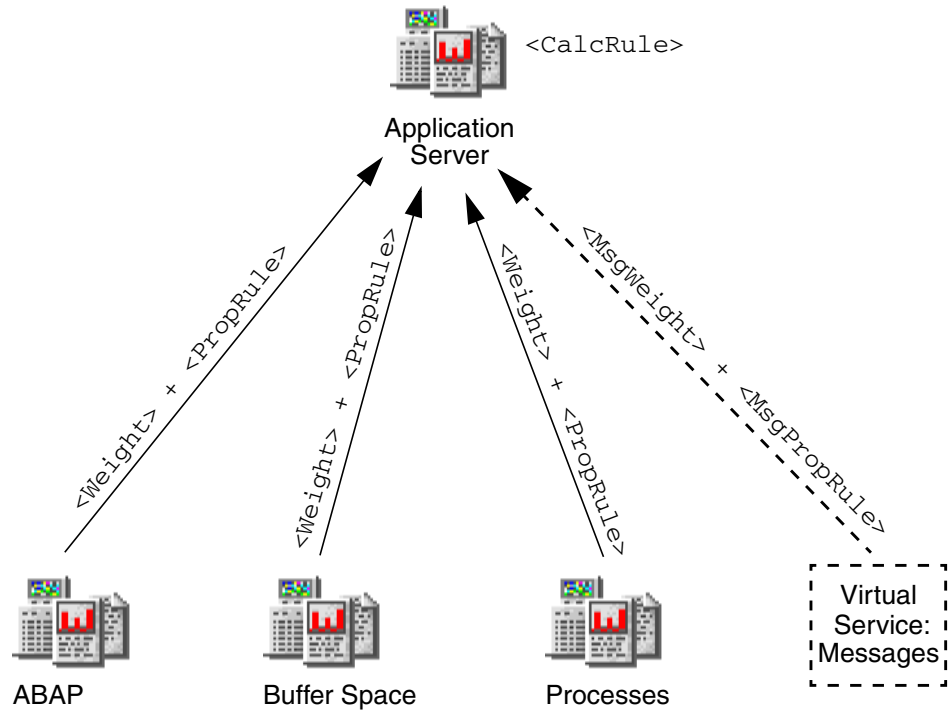
Weighting

Weighting enables you to associate a factor with a subservice. This is done by specifying the tag `<Weight>` when defining the service in the service configuration file. The subservice is then weighted more compared to all other subservices, and consequently affects status calculation differently than without the weighting factor. `<Weight>` can be a decimal number.

You can also add a weighting factor to the messages directly targeting the service. The tag is `<MsgWeight>` and is specified when defining the service in the service configuration file. `<MsgWeight>` can be a decimal number. Figure 1-7 on page 38 illustrates how the messages directly targeting the service contribute to its status. These messages can also be seen as a “virtual message service”.

Figure 1-7 on page 38 also shows that the calculation rule of the service considers both the `<Weight>` (or `<MsgWeight>`) and the `<PropRule>` (or the `<MsgPropRule>`).

Figure 1-7 Virtual Message Service



For example, if you define the weight of the subservice OS to be 2 instead of 1 which is the default for all subservices, OS is weighted more when the status of the service is calculated. However, the weight is not taken as an absolute number but as a fraction of all subservices and messages. Service Navigator calculates the factor using the following formula:

$$\text{Weighting Factor} = \frac{\text{Factor of a subservice/message}}{\text{Sum of all factors of all subservices/messages}}$$

The following example assumes that OS has been attributed the factor 2 whereas all other subservices have the default factor 1. In addition, the messages targeting the service directly are weighted with the message factor 2. The sum of all factors is 8. So according to the above formula the weighting factor for ABAP is: 1 divided by 8 equals 0.125; and the weighting factor for OS is: 2 divided by 8 equals 0.25. The message weighting factor is: 2 divided by 8 equals 0.25. See also Table 1-2.






Table 1-2

Example: Calculating the Weighting Factor

| Subservices | Weight | Factor |
|--------------------|---------------|---------------|
| ABAP | 1 | 0.125 |
| Buffer Space | 1 | 0.125 |
| Processes | 1 | 0.125 |
| Display Queue | 1 | 0.125 |
| OS | 2 | 0.25 |
| Message | 2 | 0.25 |
| Total | 8 | 1 |

When calculating the status of a service, Service Navigator multiplies each severity with the factor and arrives at the following results; see Table 1-3.

Table 1-3 Example: Status Calculation modified by the Factor

| Sub-services | Severity as interpreted by the service Application Server | Multiply with the factor |  critical |  major |  minor |  warning |  normal |
|--|---|--------------------------|--|---|--|---|--|
| ABAP | minor | 0.125 | 0 | 0 | 0.125 | 0.125 | 0.125 |
| Buffer Space | critical | 0.125 | 0.125 | 0.125 | 0.125 | 0.125 | 0.125 |
| Processes | major | 0.125 | 0 | 0.125 | 0.125 | 0.125 | 0.125 |
| Display Queue | minor | 0.125 | 0 | 0 | 0.125 | 0.125 | 0.125 |
| OS | major | 0.25 | 0 | 0.25 | 0.25 | 0.25 | 0.25 |
| Messages targeting Application Server: severity minor ^a | | 0.25 | 0 | 0 | 0.25 | 0.25 | 0.25 |
| Total | | 1 | 0.125 (12%) | 0.50 (50%) | 1 (100%) | 1 (100%) | 1 (100%) |

a. The messages directly targeting a service are seen as an entity: the message with the highest severity effects the service.

If we now compare the results with Table 1-1 on page 35, we can see the effect the weighted subservice OS and the weighted messages have on the overall status calculation. If one of the following were configured:

- ❑ **Most Critical:** the service changes to critical because the highest severity found amongst the subservices (critical) is adopted by the service.

This is not different from the example without the weighting factor on page 35 because critical is still the highest severity found amongst the subservices and messages.

- ❑ **Single Threshold 50%:** the service changes to major because the severity that first exceeds the threshold of 50% is adopted by the service. In this case the severity major with 50% is the first to exceed the threshold of 50%.

This is not different from the example without the weighting factor on page 35 because major is still the first to exceed the threshold.

- ❑ **Multiple thresholds:**
 - threshold of 15% for severity critical
 - threshold of 50% for severity major
 - threshold of 60% for severity minor
 - threshold of 70% for severity warning

The service changes to major because the first threshold to be exceeded is 50% (severity major). Note the difference here to the example without the weighting factor on see page 35: critical does not exceed the threshold but major does. So instead of adopting the severity critical the service adopts the severity major.

Note that there is no need to specify a threshold for the severity normal because normal is the lowest severity status.

How Does OVO Map Messages to Services?

OVO generates a message as a result of an event or change of status on a managed node. If the event matches preconfigured filters (that is, the OVO message conditions), a message is generated, formatted, and then sent to the responsible user. If the event does not match the filtering criteria, the event is ignored.

Formatting takes place by attaching attributes to the message. Attributes are, for example, the node or application on which the event happened, the object that was affected, or the message group of the message.

Service Navigator enables you to attach a **service name** as an additional attribute to the message. If an event occurs on a managed object and this event matches the filtering criteria, OVO generates a message and attaches the service name as an attribute to this message. The service name of the message is then compared with the service names in your hierarchy. If the service name of the message is matched by a service name in the hierarchy, the message is sent to the GUI of the user who is assigned to that service.

When mapping messages to services, you should first consider what the problem is, and then which service is affected by it, before setting the service name.

The service name can be set in one of the following ways:

Template and condition windows

Message mapping takes place on the node that generated the message.

Regroup Condition No. window

Message mapping takes place on the management server.

opcmsg(1) parameter

For example, `/opt/OV/bin/OpC/opcmsg application=appl \ object=obj msg_text=msg_text service_id=service_id`

Naming the service takes place when you define the service in the configuration file. Service names must be unique for each service so that OVO can identify the service and map messages to that service. Note that the service name you enter in the OVO administrator GUI can contain OVO variables and can make use of OVO's pattern-matching mechanism. This has the advantage that you can keep your OVO templates as generic as possible, namely by using variables to match service names, and that you can use one template to match several services.

In which situation should you set the service name in message conditions or in message regroup conditions?

Message conditions

- You can take advantage of the filtering mechanism of OVO conditions.
- You can use more variables; the message conditions offer more variables than message regroup conditions.
- You are nearer to the instrumentation; for example, HP OpenView Smart Plug-ins use message conditions.

Regroup conditions

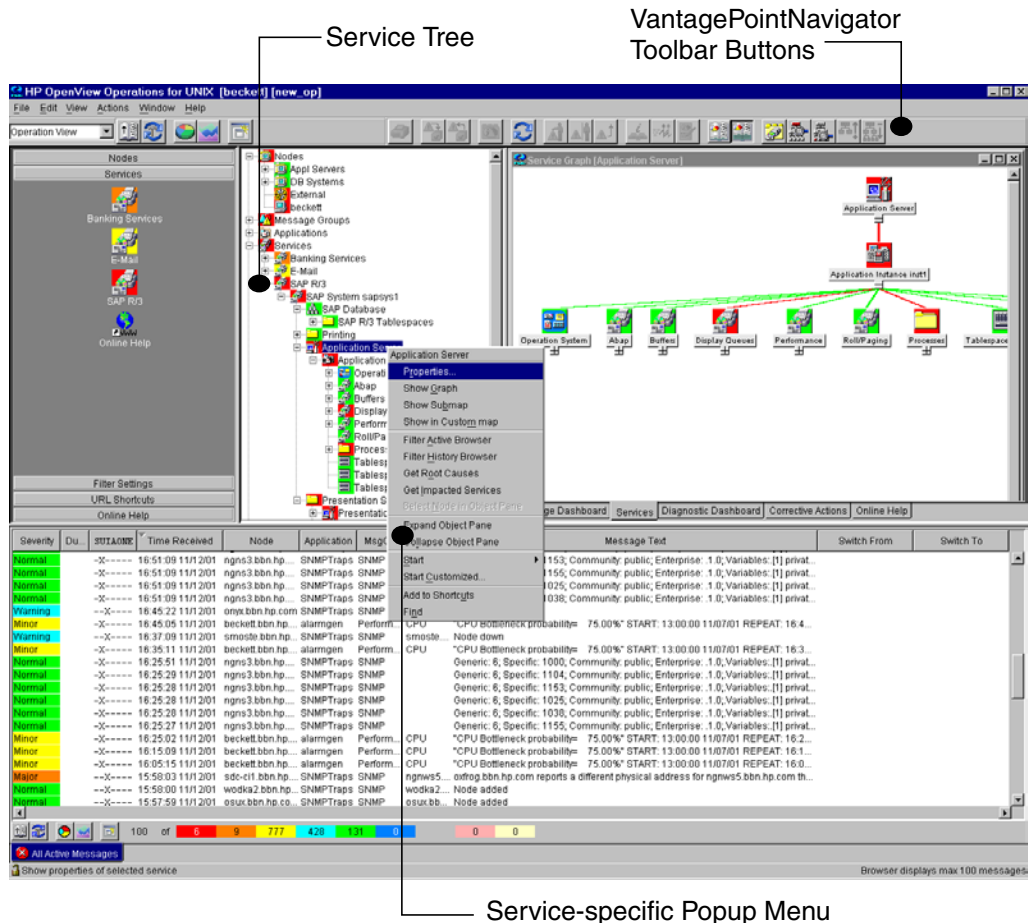
- You want a quick solution rather than having to change a large number of templates.
- You want to avoid distributing templates.

The Service Navigator GUI

When you start the Service Navigator GUI you see that the OVO Java-based GUI is extended to include the following:

- ❑ a service tree in the scoping pane on the left.
- ❑ additional toolbar buttons.
- ❑ additional options in the menu bar and popup menu.

Figure 1-8 The Service Navigator GUI

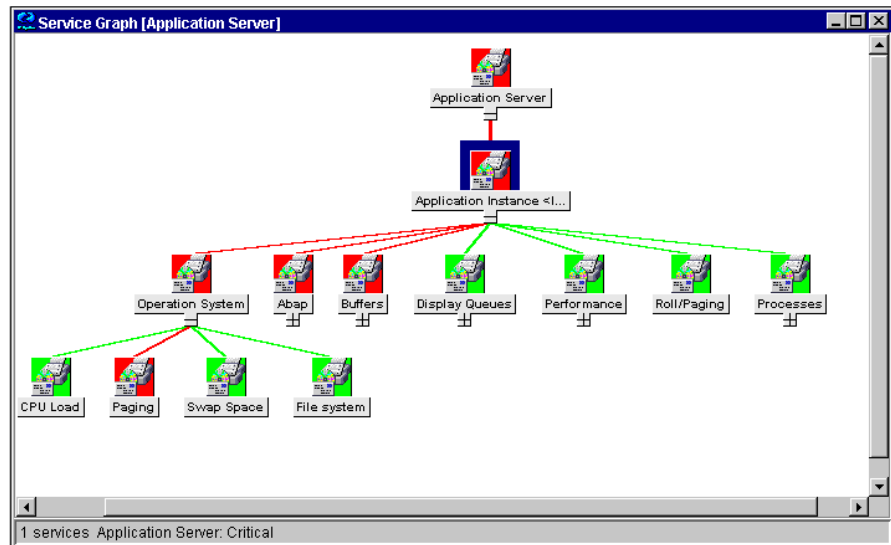


Service Graphs

Service Navigator displays a simple hierarchical view of services through a service tree in the scoping pane of the GUI. This information can also be graphically presented in the content area with service graphs. These are graphical representations of the hierarchy of services and their subservices. Service graphs are dynamic and are updated whenever the severity of a displayed service changes. An example of a service graph is shown in Figure 1-9.

Figure 1-9

A Service Graph



TIP

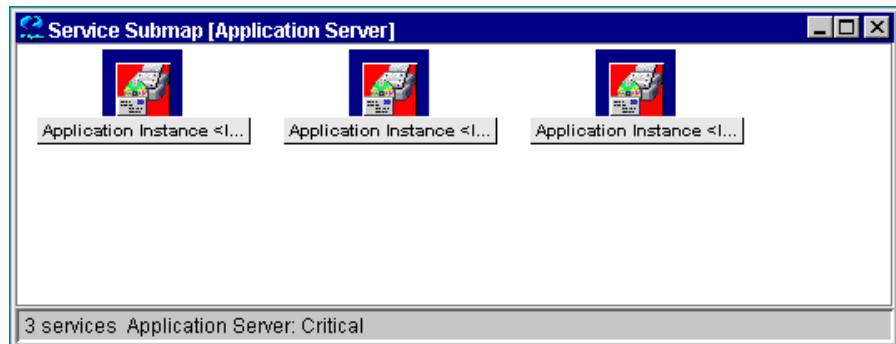
You can also double-click a service icon in the service graph to expand or collapse the service hierarchy below. If this service's submap is defined to be displayed with the background image display, it will be opened instead.

Service Submaps

Service submaps are graphical representations of subservices of a selected service. No hierarchical information is displayed. Service submaps are most useful for providing a clear view of the status of the subservices within parent services when hierarchical relationships are not of primary interest. An example of a service submap is shown in Figure 1-10.

Subservice submap of the current service submap are accessed by double-clicking the service icon.

Figure 1-10 A Service Submap



Enhancing Service Submaps

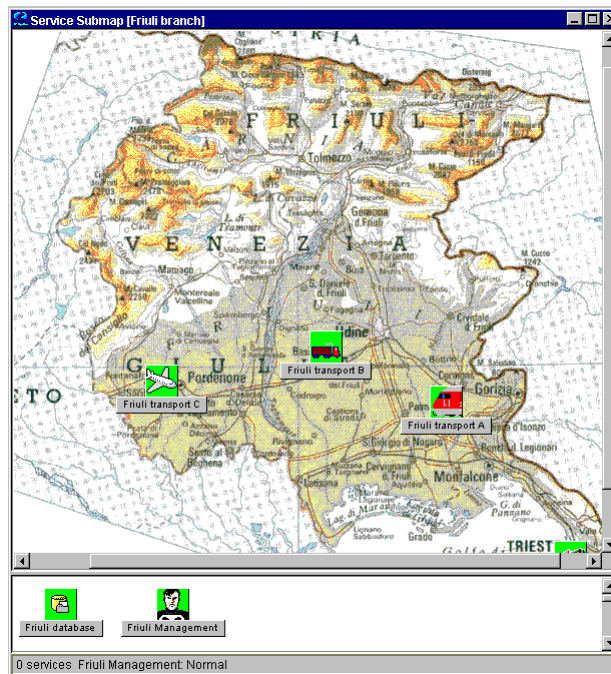
Service submaps can be enhanced with alternative background images, for example a network schematic or a country map. You can also define the display position of subservice icons on the background image to create clearer overview of the services being managed. For example, position service icons over their geographical locations on a country map.

The customized Service Submap window of a service may be split into two horizontal panes, when there are services which have not been allocated any positional information. These icons are then placed in the lower pane.

Figure 1-11 shows the service submap of the Friuli branch service. The Friuli branch service has some subservices located on the submap background and others without positional information placed in the lower pane.

Figure 1-11

An Enhanced Service Submap



For information on how to customize your service submaps, see “... How to Enhance a Service Submap?” on page 128 for details.

Custom Service Maps

Custom service maps are graphical representations of services. Multiple services from the service hierarchy can be selected and displayed within a custom service map. Custom service maps are very useful for maintaining an overview of the most important, top-level services or even displaying them on a large screen in the IT Center.

Service Properties

Service properties contain detailed information about a service. This information can be viewed in the Service Properties dialog box. The following information is displayed in different tabs:

□ General

The General tab of the Services Properties dialog box includes basic information about the selected service, such as service name, display label, a short description of the service, node to which the service is attached, and the current status of the service.

□ Actions

The Actions tab of the Services Properties dialog box displays the actions assigned to the selected service and includes the unique name, the type, and a detailed description of the service action.

□ Attributes

The Attributes tab of the Services Properties dialog box displays the attributes assigned to the selected service. Service attributes are defined by the administrator and can be used to convey any additional information about a service.

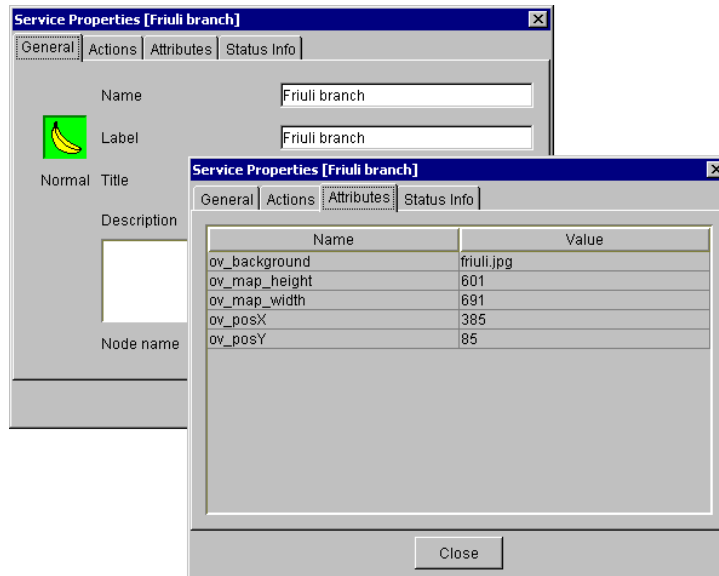
Reserved attributes are used to apply background images to service submaps, position subservice icons on these backgrounds, and add additional labels to service icons. For more information, refer to “Reserved Service Attributes” on page 165.

□ Status Information

The Status tab of the Services Properties dialog box displays parameters and their values for the selected service. For example, a parameter of a service could be the number of messages for the service and its current value could be 12.

Figure 1-12 displays the general properties and service attributes tabs of a service.

Figure 1-12 **Displaying the Properties of a Service**



Labeling Services

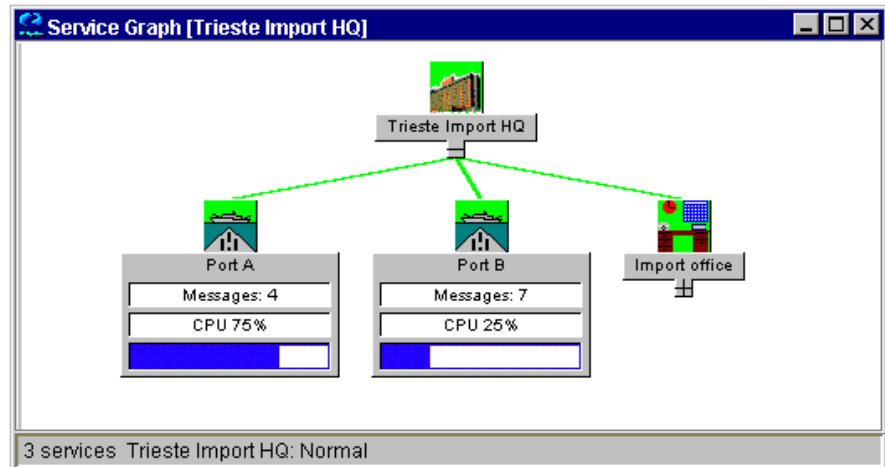
All service icons are identified using a label. The label appears below an icon in the shortcut bar, service graph, service submap, and custom service map. Additional text or image labels can also be displayed by service icons. An unlimited number of labels can be applied to an icon, the practical limitation being the space available on the display.

Additional labels are used to display information that is important for monitoring the service. For example, the CPU load of the systems that are hosting the service can be displayed as a numerical value and as a bar graph for at-a-glance information. An example of how these labels can be applied to service icons is shown in Figure 1-13.

A collection of reserved service attribute names that are recognized by the GUI are used for labeling. For more information, refer to “Reserved Service Attributes” on page 165.

Figure 1-13

Labeling a Service



There are two ways of labelling services using reserved service attributes:

❑ **Service Configuration File**

As part of the service configuration file. The values for these labels are acquired at Service Navigator start-up and cannot be changed or removed during the current session.

❑ **Dynamically**

It is also possible to automatically update the text or image of a label and display the most up-to-date information by the use of messages. The refresh rate is determined by that set for the message browser.

For more information, refer to “... How to Label Service Icons?” on page 124.

Loading Services on Demand

Loading a large service tree containing hundreds of services can take a very long time. The OVO Java GUI enables you to set the number of levels in a service hierarchy to be retrieved from the management server. This way the time needed for Java GUI startup is significantly reduced.

It is also possible to configure an automatic reload of the service configuration when a change occurs on the management server, so that services information is regularly updated. You can choose to retain only the services that are displayed in service graph or Object Pane, visible or minimized, in the Java GUI cache. All other retrieved service information is removed from Java GUI cache.

For more information about loading services on demand, refer to the *OVO Java GUI Operator's Guide*.

Service Logging and Reporting

A critical aspect of IT management is the capability to generate reports about your service delivery performance.

If configured, Service Navigator creates a log entry in the OVO database whenever a service changes severity. Reports retrieve the logs from the OVO database and display the information in graphical or statistical form. You can design and generate your own reports or use the service reports that are bundled with HP OpenView Reporter.

Service reports cover a wide range of reports, starting from reports about the current status of a service, to reports about trends based on historical status data. The following Service Navigator reports are bundled with Reporter:

❑ **Service availability**

This report displays the duration of a service in each status. Each status is presented in percent using a pie chart. Alternatively, a bar chart can be displayed. The report is based on all status logs in the OVO database.

❑ **Message trend by service**

This report displays the number of messages received for each service over time. It is based on all active and history messages in the OVO database that affect a service. The report displays a line graph for each service.

❑ **Top active messages by service**

This report displays the three services with the highest number of active messages, broken down by severity. The report is based on all active messages in the OVO database and displays a bar chart.

See the HP OpenView Reporter documentation for more information about these reports. For information about the contents of the OVO database and how to go about designing and generating your own reports, see the *OVO Reporting and Database Schema*.

Service logging is easily enabled (or disabled) using a Service Navigator command line tool. Command line tools are also available to download or upload service data from or into the OVO database. This can be useful for database maintenance purposes. See “Enabling Service Logging” on page 74 for more information.

Note that the OVO Developer’s Toolkit includes APIs for Service Navigator to get the current status of a service and to register for service status changes. See the *OVO Application Integration Guide* for more information.

The service logging file lists the services for which logging is enabled and is located in:

```
/etc/opt/OV/share/conf/OpC/mgmt_sv/opcsvcm/loggings
```

NOTE

Never edit this file directly.

2 **Getting Started**

In This Chapter

This chapter answers the following questions:

- ❑ How Do I Install Service Navigator?
- ❑ How Do I Configure Service Navigator?
 - Planning Your Service Hierarchy
 - Writing the Service Configuration File
 - Activating the Service Configuration
 - Setting the Service Name in OVO
 - Assigning Services to Operators
 - Planning Your Service Hours
 - Enabling Service Logging
 - Setting Service Attributes Dynamically
- ❑ How Do I Start Service Navigator?
- ❑ How Do I Upgrade Service Navigator?
- ❑ How Do I De-install Service Navigator?

How Do I Install Service Navigator?

Installing Service Navigator

To ensure that the Service Navigator is properly installed on your management server system, perform the following:

❑ **Adapt kernel parameters**

The kernel parameter `max_thread_proc` must be set appropriately. See “Determining the size of `max_thread_proc`” on page 60.

❑ **Install OVO**

OVO A.08.10 management server as described in Chapter 2 of the *OVO Installation Guide for the Management Server*.

IMPORTANT

Service Navigator is installed automatically during the installation of OVO A.08.10 management server.

❑ **Install the Java-based Operator GUI**

OVO A.08.10 Java-based Operator GUI as described in Chapter 3 of the *OVO Installation Guide for the Management Server*.

NOTE

If you have a previous version of Service Navigator installed and configured, see the section “How Do I Upgrade Service Navigator?” on page 94 for more information about upgrading to the current version of Service Navigator.

Determining the size of max_thread_proc

The kernel parameter `max_thread_proc` determines the maximum number of threads allowed in each process. By default, OVO and Service Navigator require this parameter be set to 1024. This value is a result of the following formula and is sufficient for standard OVO and Service Navigator environments. For larger environments, use the following formula as a guideline for increasing this value.

$$\begin{aligned} \text{threads} = & \quad [NumberOfGUIs * 2] + \\ & \quad [NumberofAPIClients] + \\ & \quad [Numberofopcservice] + x + 9 \\ & \quad x = 1 \text{ (if service logging is enabled)} \\ & \quad x = 0 \text{ (if service logging is disabled)} \end{aligned}$$

where:

NumberOfGUIs =

Number of running Service Navigator GUIs.

NumberofAPIClients =

If using the APIs of the Developer's Toolkit:

Number of API clients.

Numberofopcservice =

Number of concurrent `opcservice` commands. `opcservice` uses internally the APIs of the Developer's Toolkit. Therefore, the number of concurrent `opcservice` commands that may run at any time must be added to calculate the value for `max_thread_proc`.

How Do I Configure Service Navigator?

Service Navigator is easily configured if you follow these step-by-step instructions:

1. Plan your service hierarchy. Identify the managed elements that are part of your service and plan your service hierarchy accordingly.
See “Planning Your Service Hierarchy” on page 62.
2. Write the service configuration file (or convert existing configuration files to the XML syntax).
See “Writing the Service Configuration File” on page 64.
3. Activate your new service configuration with `opcservice`.
See “Activating the Service Configuration” on page 66.
4. Set the message attribute service in OVO. Configure OVO so that it knows how to assign messages to services.
See “Setting the Service Name in OVO” on page 67.
5. Plan the operator responsibilities and assign them with `opcservice`.
See “Assigning Services to Operators” on page 71.
6. Plan your service hours. When do you provide your service?
See “Planning Your Service Hours” on page 73.
7. Plan the reports you need for each service and enable service logging accordingly.
See “Enabling Service Logging” on page 74.

See also Chapter 3, “The Service Configuration File,” on page 99 for more detailed instructions for writing the service configuration file.

NOTE

Some HP OpenView Smart Plug-ins provide out-of-the-box service instrumentations.

Planning Your Service Hierarchy

We recommend that you draft your service hierarchy before you start writing the service configuration file. When planning your service hierarchy, keep the following questions in mind:

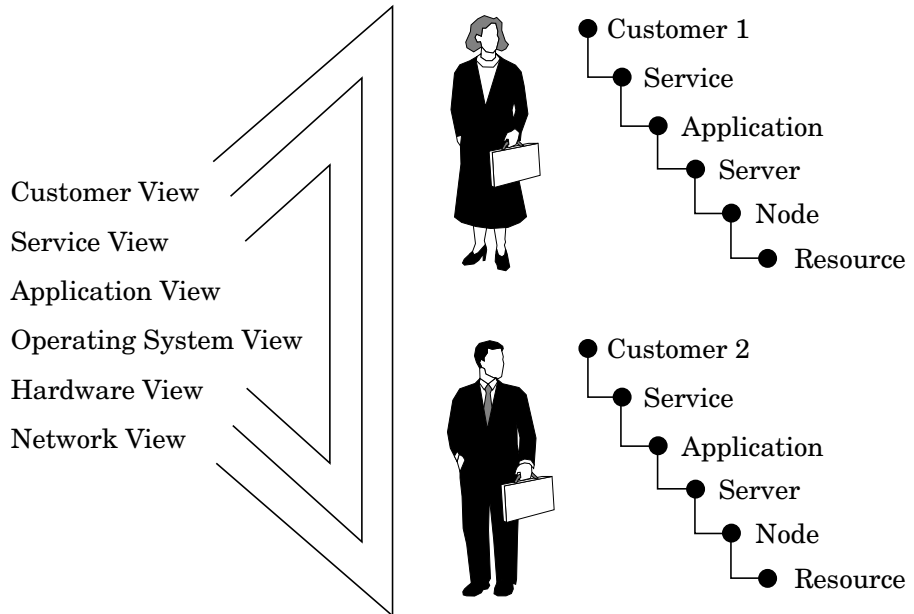
- Which IT services do you provide? Which ones do you want to monitor?
- Who are the customers of your services? Which organizations, departments, or lines of business?
- How can you logically group the services you provide? Which services are used by other services?
- How do problems in one service affect another? Which status propagation rule should you apply?
- How do you evaluate the severity of a problem? Which status calculation rule should you apply?
- Which service actions should be assigned to each service?

For example, you may want to organize your service hierarchy as illustrated in Figure 2-1 on page 63: your service hierarchy could start with the customers for which you provide services, followed by the business services you provide for them, for example, a backup service. The next level could be the application level; in the case of your backup service, this could be the HP OpenView OmniBack II application you are using for backup purposes at your customer sites. The following level could be the software or operating system level where you monitor the health of the operating system installed on your backup servers. The next level could be the hardware level where your servers and clients are monitored. Finally, the last level could be the network level which enables you to control the health of all network components.

NOTE

Only ASCII characters are allowed in service names.

Figure 2-1 Service Hierarchy Example



See also “... That Names are Important?” on page 115 for more information about planning a service hierarchy.

You can also have a look at the example service configuration files. They show you what a service hierarchy can look like and how to use the XML tags. The example configuration files are available in the following directory on the management server after the installation of Service Navigator:

```
/opt/OV/OpC/examples/services/
```

See the README file for more information about each file.

Writing the Service Configuration File

When you have a good plan of your service hierarchy, you can start putting it into practice by writing the service configuration file. In general, you have the choice between the following methods:

Manually

Manually write (or edit) a service configuration file as explained in this section.

Automatically

Automatically generate a service configuration file using scripts or programs. Service Navigator provides some shell scripts in the directory `/opt/OV/OpC/examples/services/` that help you to get started.

Use the example configuration files to copy the sections of the configuration syntax that you need. See also Chapter 3, “The Service Configuration File,” on page 99 for information about how parts of the SAP/R3 service hierarchy have been configured.

TIP

File Name Extensions

Save the service configuration files with the standard file name extension `.xml`.

Large Service Hierarchies

If you are planning a large service hierarchy, it can be useful to distribute the service configuration over multiple configuration files; this would make it easier for you to maintain your configuration. For example, you could define your business services and application services in one file, and all hardware and software services in a second file.

1. Change to the directory containing the example configuration files.
Enter:

```
cd /opt/OV/OpC/examples/services/
```

2. As user `root`, copy the file that comes closest to your planned hierarchy to a temporary directory. Enter:

```
cp <service_example> /tmp <service_config>
```

3. Open the copied example file using an XML or ASCII editor of your choice and translate the service hierarchy you have planned into the service configuration syntax. See “The Service Configuration File Syntax” on page 143.

Configuring a service includes the following aspects:

- a. Defining the services and subservices, and the relationships between them.
 - b. Defining the status calculation and status propagation rules.
 - c. Defining the service-specific actions.
 - d. Defining the service attributes.
 - e. Defining operator assignments.
4. Use the `opcservice` command on the finished configuration file to ensure that your changes are correct:

```
/opt/OV/bin/OpC/opcservice -check /tmp/<service_config>
```

If the syntax of the configuration file is correct, `opcservice` will output an OK message. If the configuration file contains any errors, errors messages are displayed. Fix any syntax errors before continuing.

If element values for service files are not specified, warning messages are displayed. To handle error messages, see “Service Files Error Checking” on page 67 for more details.

Activating the Service Configuration

When you have finished writing the service configuration file, use the command `opcservice` to load the new configuration. Enter:

```
/opt/OV/bin/OpC/opcservice -add /tmp/<service_config>
```

`opcservice` passes the configuration to the `opcsvcm` process which maintains the internal service repository file:

```
/etc/opt/OV/share/conf/OpC/mgmt_sv/opcsvcm/services
```

NOTE

You should not edit this file directly. Any changes done during runtime are lost the next time the `opcsvcm` process rewrites the repository file.

Modifying the Service Configuration

If you need to modify the configuration, execute the following with the management server running:

```
opcservice -list -all -xml > current_ServNav.xml
```

```
cp current_ServNav.xml new_ServNav.xml
```

Edit the new configuration file, remove the `<Results>` and `</Results>` tags and update the configuration with the following command:

```
opcservice -add new_ServNav.xml
```

NOTE

Do not use the `-replace` option, otherwise you will have to re-do the operator assignments.

The execution of `opcservice -list -all -xml` is time consuming.

You can also get an overview of your configuration by listing all services with

```
opcservice -list -all.
```

See also “The `opcservice` Command” on page 168 for more information about available command line options for `opcservice`.

Service Files Error Checking

OVO implements error checking of service files. If element values are *not* specified, by default, warning messages are displayed. For example:

```
Warning: Operation 'Add' : Element 'car_manufact' Label is empty,  
using Element name instead. (SVC50-3) (SVC10-123)
```

```
Warning: Operation 'Add' : Element 'supply-chain' Label is empty,  
using Element name instead. (SVC50-3) (SVC10-123)
```

To enable error checking, enter the following command:

```
ovconfchg -ovrg server -ns opc -set \  
OPC_SVCM_ADD_WARN_IF_EXISTS <mode>
```

Where *<mode>* is one of the following:

- TRUE
- FALSE

IMPORTANT

If you are using OVO version earlier than A.08.10, you can enable error checking by editing the `opcsvinfo` file. Enter the following line:

```
OPC_SVCM_ADD_WARN_IF_EXISTS <mode>
```

Where *<mode>* is either TRUE or FALSE.

Setting the Service Name in OVO

When you have set up your services, you need to tell OVO how to match messages against services. You can either set the message attribute `Service Name` in the message source template or condition, in a message regroup condition, or as a parameter of `opcmsg(1)`.

Note that you can enter variables as shown in Figure 2-3 on page 70. See “... How to Use OVO Variables?” on page 118 for more information.

Message source template or condition

Setting the message attribute `Service Name` when the message is generated on the managed node:

1. Select Window: Message Source Templates from any main OVO window. The Message Source Templates window opens.

2. Depending on whether you want to enter the name of your service in a template or condition window, modify the template or condition. If you set the service name in a message source template, it becomes the default value for all subsequent conditions.

Note that you can use OVO variables as part of the service name in the *Service Name* field. This generates the service name from the message attributes. See “OVO Variables” on page 173 for more information.

3. Distribute the new configuration to the managed nodes.

See Figure 2-2 on page 69 for an example of a *Condition No.* window.

Figure 2-2 Condition No. Window

Condition No. 1

Description
Failed remote login

Condition

Node
[]

Message Text
FAILED <@.user> <@.tty> <@.host> <@.date> <@.time>

- Suppress Matched Condition
 = Suppress Unmatched Condition
 + Message on Matched Condition

Advanced Options...

Set Attributes

| Severity | Node | Application | Message Group | Object |
|-----------|------|-------------|---------------|--------|
| unchanged | [] | [] | [] | <user> |

Message Text
Failed login of <user> on <tty> from <host> at <time> <date>

Service Name
SAP

Message Type
[]

Custom Attributes... Instructions... Message Correlation...

Actions

On Server Log Only (put directly into History Log)

| | Node | Command | Anno. | Ackn. |
|--------------------|------|---------|-------|-------|
| Automatic | [] | [] | No | No |
| Operator Initiated | [] | [] | No | No |

Forward to Trouble Ticket
 Notification

OK Cancel Test Pattern Matching... Help

Service Name

❑ Regroup condition

Regrouping services on the management server:

1. Select **Actions:Server->Regrouping...** from the menu bar of the OVO Node Bank. The Regroup Conditions window opens.
2. In the Regroup Conditions window click [Add...] and enter your new service in the New Service Name field in the Regroup Condition No. window. See Figure 2-3 for an example of a Regroup Condition No. window.

The service messages matching this regroup condition are forwarded to the service you have specified.

Figure 2-3 Regroup Condition No. Window

Service Name

❑ `opcmsg()`

You can also provide the service name without using the OVO GUI by specifying it as a parameter of the `opcmsg(1)` command. See the `opcmsg(1)` man page for more information.

Assigning Services to Operators

A service is only visible to the operator or operators to whom you have assigned responsibility for that service. If a service is assigned to more than one operator, its status is visible to all operators.

Note that the operators should also have message groups and node groups assigned. If responsibilities for services, nodes, and message groups do not overlap, the operators can receive messages from a service but do not have the corresponding nodes in their responsibility matrix. Messages from assigned services are also displayed in the Motif-based operator GUI so that an operator who uses the Motif GUI will also see messages from assigned services.

Operator capabilities are not restricted on service messages or on the nodes from which service messages are received. For example, if an operator is in general not allowed to own/disown messages, messages from assigned services can still be owned or disowned. Similarly, actions can be executed on service nodes, even if these nodes are not in the responsibility matrix of the operator.

NOTE

User profiles are not supported. You have to assign services to each operator separately.

Services can be assigned using one of the following methods:

❑ **opcservice command**

Use the `opcservice` command to assign services to operators. This is the recommended method.

❑ **service configuration file**

Define the assignment in the service configuration file using the `<Operator>` tag.

If you want to find out which operators are responsible for your services, use `opcservice` with the `-operators` option. This outputs a list of all operators who have services assigned as well as the assigned services. Service assignments are kept in the OVO database.

Assigning Services with opcservice

1. Decide which operator is responsible for a service.

Note that in a complex service hierarchy it may be useful to assign subservices to different operators instead of the full hierarchy.

2. Use `opcservice` to assign the service and all contained subservices to the operator. Enter:

```
/opt/OV/bin/OpC/opcservice -assign <operator> \  
<service_name>...
```

Where:

| | |
|-----------------------------------|---|
| <code><operator></code> | OVO login name of the operator. |
| <code><service_name></code> | Name of the service as defined in the service configuration file. Multiple service names can be listed. |

See also “The `opcservice` Command” on page 168 for more information about available command line options for `opcservice`, for example for deassigning services from operators.

Assigning Services in the Configuration File

Service assignments can also be specified in the service configuration file. This involves defining the operators with the XML tag `<Operator>` and specifying the desired services.

This method has the advantage that you can define additional aspects of the Service Navigator GUI, for example, you can define the label, description, or icon of the topmost service assigned to the operator. Note that using `opcservice` is the recommended method. See also “Setting up Service Assignments” on page 111.

Planning Your Service Hours

With OVO you can set up service hours and schedule outages. Messages which arrive outside of defined service hours are buffered, and messages which arrive during a scheduled outage are suppressed. See the *OVO Concepts Guide* and the *OVO Administrator's Reference* for more information.

Enabling Service Logging

If enabled, Service Navigator keeps a log of each status change in the OVO database. This enables you to design and generate reports about the availability of your services based on the data kept in the OVO database.

1. Decide for which services you want to enable logging.
2. Enable logging, enter:

```
/opt/OV/bin/OpC/opcservice -log_enable <service_name>
```

To enable logging for all subservices or for subservices up to a certain level, use the `-recursive` or `-level` options respectively.

3. To check for which services logging is enabled, enter:

```
/opt/OV/bin/OpC/opcservice -logs
```

See also the man page *opcservice(1M)* for more information about disabling service logging and other command line options.

The service logging file lists the services for which logging is enabled. It is located in:

```
/etc/opt/OV/share/conf/OpC/mgmt_sv/opcsvcm/loggings
```

NOTE

Never edit this file directly.

Use the command line tools *opcsvcdwn* and *opcsvcupld* to download or upload service logs from or into the OVO database. Downloading service logs can become necessary when too many logs exist in the OVO database and the amount of available disk space is not sufficient. See also the man pages *opcsvcdwn(1M)* and *opcsvcupld(1M)* for more information about these commands. More details about service logging and the service-related tables are also available in the *OVO Reporting and Database Schema*.

Setting Service Attributes Dynamically

Service attributes are used to provide additional information about a service. You can define service attribute values in the service configuration file. Information on defining service attributes is available in “Setting Service Attributes” on page 109. Service attributes that are specified in the service configuration file are loaded at start-up and cannot be updated during runtime.

Attribute values can also be added dynamically without the need to update the service configuration file and restart Service Navigator. This has the advantage that Service Navigator is always updated with the latest service attribute values and you are always presented with the latest service information. You can also dynamically update service attribute values, which are already specified in the service configuration file. Any values which are dynamically set can also be stored and reloaded the next time Service Navigator is started by enabling the recovery feature.

There are two ways to set service attributes dynamically:

❑ **opcsvcattr command line tool**

`opcsvcattr` is a shell script that can be called on the command line to add, change, or remove service attributes.

This script is useful for testing purposes to verify that the service attributes are updated as expected. You can also call this tool as an automatic action for a message. Beware though that performance could be impaired.

Logging is not available for this method.

To find out how to use the `opcsvcattr` tool, refer to “Setting Service Attributes Dynamically with `opcsvcattr`” on page 77.

❑ **OVO messages**

OVO message attributes can be used to convey updated information used to dynamically set service attribute values. The following message attributes are used:

- *Application and object message attributes*

This method is available for all message sources.

- *Custom message attributes*

This method is only available for message sources where custom message attributes are supported: logfiles, opcmsg(1 | 3) and threshold monitor.

Using OVO messages to set service attributes has the advantage that performance is not impaired and that the changes can be logged. However, the configuration effort is higher than when using the `opcsvcattr` tool.

See “Setting Service Attributes Dynamically with OVO Messages” on page 79 for more information about this method.

What to be Aware of When Setting Service Attributes Dynamically

Attributes are added at runtime and when the Service Navigator is stopped, the values are deleted unless message logging is enabled. A service attribute that is added dynamically overrides the value of the service attribute of the same name. When the service configuration is reloaded, the service attribute value that is defined in the service configuration file will again be displayed unless message logging is enabled.

To find out how to enable logging, see “Message Logging for `opccustproc1`” on page 85.

TIP

You can save a current service configuration to a file in XML format with the `opcservice -dump <filename>` command. Portions of this configuration can be used to enhance the associated service configuration file.

Setting Service Attributes Dynamically with opcsvcattr

Use the shell script `opcsvcattr` to add, change, or remove service attributes dynamically. For example, to add the three labels, `Master`, `Low`, and `10.5V`, to the service `ABANO`, enter the following command on the management server:

```
/opt/OV/bin/OpC/opcsvcattr svc_id=ABANO \  
name=ov_label11 value=Master\  
name=ov_label12 value=low\  
name=ov_label13 value="10.5V"
```

The icon representing the service `ABANO` now displays three new labels with the values as illustrated in Figure 2-4.

Figure 2-4 Example Results for `opcsvcattr`



TIP

Service Navigator uses reserved attribute names such as `ov_label11`, `ov_label12`,... `ov_labeln`, to add additional labels to service icons. See the section “... How to Label Service Icons?” on page 124 and “Reserved Service Attributes” on page 165 to learn more labeling service icons.

The `opcsvcattr` tool can be used as an automatic action in the OVO templates. Beware though that this may cause performance problems. Ideally, `opcsvcattr` should only be used when testing the configuration of service attributes.

Syntax for `opcsvcattr`:

```
opcsvcattr  svc_id=<svc_name> {name=<name> value=<value>} |
            -remove svc_id=<svc_name> {name=<name>} |
            -removeall svc_id=<svc_name> |
            -removepref svc_id=<svc_name>name=<name_prefix>,
```

| | | |
|--------|----------------|--|
| where: | <svc_name> | Name of the service to which the service attributes apply. |
| | <name> | Service attribute name. For example <code>ov_label1</code> . |
| | <value> | Service attribute value to be displayed. |
| | -remove | Removes all specified service attributes. |
| | -removeall | Removes all service attributes. |
| | -removepref | Removes all service attributes which name matches the specified name prefix. |
| | <name_ prefix> | Prefix string for -removepref. |

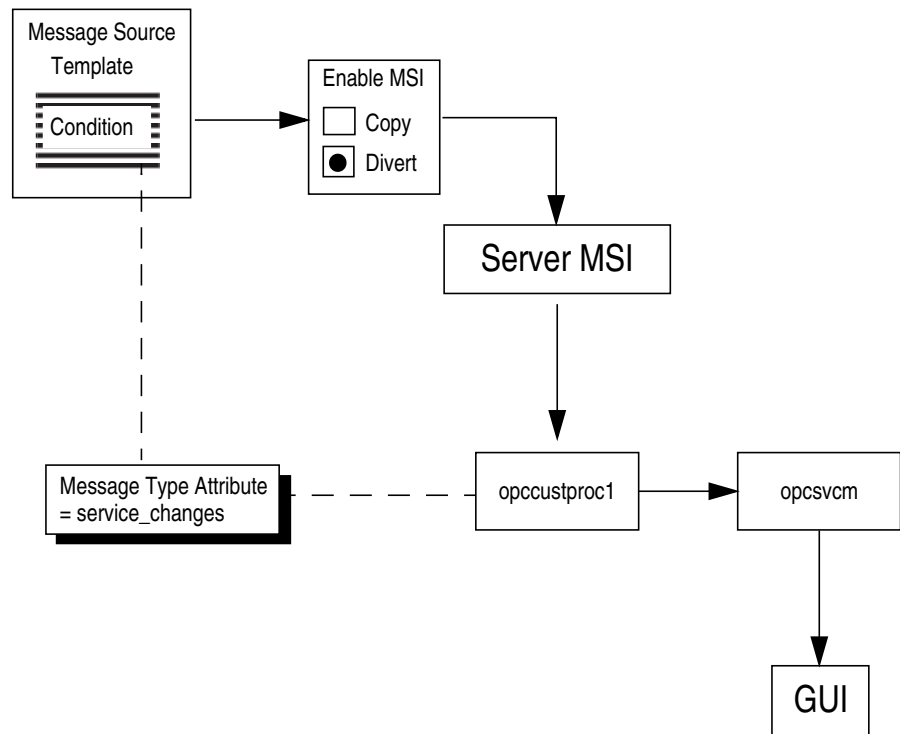
For more information about `opcsvcattr`, refer to the *opcsvcattr(1M)* man page.

Setting Service Attributes Dynamically with OVO Messages

Use OVO messages to transport service attributes from managed nodes to the management server. Messages that contain information relevant for service attributes are fed into the server message stream interface; the message flow of the management server. On the management server, the process `opccustproc1` retrieves those messages from the flow and, using the `opcsvcm` process, changes or removes the service attributes in the Service Navigator GUI.

See Figure 2-5, “Message Flow with `opccustproc1`,” for an illustration of the message flow from the managed nodes to the Service Navigator GUI through the management server.

Figure 2-5 Message Flow with `opccustproc1`



Information for service attributes can be attached in two ways to OVO messages. For both ways, message attributes are used to convey relevant information:

❑ **Application and object message attributes**

This method is available for all message sources.

❑ **Custom message attributes**

This method is only available for message sources where custom message attributes are supported: logfiles, opcmmsg(1 | 3) and threshold monitor.

To set service attributes:

1. On the management server, enable the process `opccustproc1`:

```
cp /opt/OV/OpC/examples/progs/opccustproc1\  
/opt/OV/bin/OpC
```

```
cp /opt/OV/OpC/examples/progs/libopccustproc1.sl \  
/opt/OV/lib
```

This integrates `opccustproc1` into the management server processes. `opccustproc1` can be controlled with the `ovstart/ovstop` or `opcsv` commands.

2. On the management server, enable output to the server message stream interface (MSI):

In the Configure Management Server window, select the option Enable Output for the MSI.

Do *not* select Send All Messages to Server MSI.

3. Prepare a message source template and condition with the following characteristics:

- a. Enable output to the Server MSI in the Message Condition Advanced Options window.

Divert the messages rather than copying them to avoid filling the message browser with irrelevant messages.

- b. Set the message type attribute to `service_changes`.

`opccustproc1` uses the message type attribute to identify the messages to be sent to Service Navigator.

- c. Depending on whether your message source supports custom message attributes or not, either set custom message attributes or the application and object message attributes:

- *Application and object message attributes*

The application and object message attributes are available for all message sources.

Enter the operation (set or remove) in the Application field and the parameters (service attribute name and value) in the Object field. See Figure 2-6 on page 81 for an illustration.

- *Custom message attributes*

Custom message attributes are only supported for the following message sources: logfile, opcmmsg(1 | 3) and threshold monitor.

You need to define two attributes: the first attribute specifies the operation (set or remove), the second attribute specifies the parameters (service attribute name and value). See Figure 2-7 on page 82 for an illustration.

See the section “Syntax for Setting Service Attributes Dynamically” on page 83 for detailed information about the required syntax.

- 4. Distribute the template to the managed nodes.

Figure 2-6 Application and Object Message Attributes for opccustprocl

| Severity | Node | Application | Message Group | Object |
|-----------|------|--------------|---------------|--------------------|
| unchanged | [] | SVC_ATTR_SET | [] | [ov_label1=Master] |

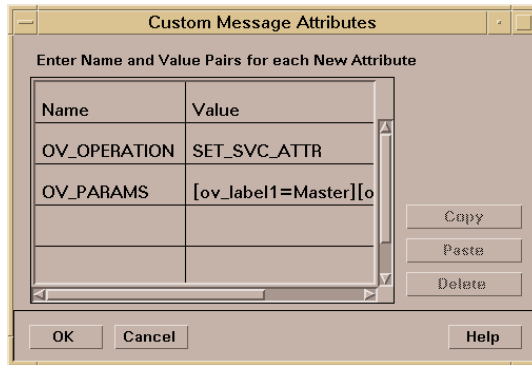
Message Text
[]

Service Name
my_service

Message Type
service_changes

Custom Attributes... Instructions... Me...

Figure 2-7 Custom Message Attributes for opccustproc1



Syntax for Setting Service Attributes Dynamically

When setting service attributes with `opccustproc1`, you always need to define two parts:

❑ Operation

The operation part instructs `opccustproc1` what to do: set or remove a service attribute.

When setting service attributes with custom message attributes, the name of the custom message attribute is `OV_OPERATION`.

❑ Parameter

The parameter part contains the data for the service attribute. It consists of a name and the corresponding value.

When setting service attributes with custom message attributes, the name of the custom message attribute is `OV_PARAMS`.

Table 2-1 on page 84 describes the syntax for custom message attributes. The `OV_OPERATION` column lists the operations that are possible (set or remove). The `OV_PARAMS` column lists the names and values for service attributes.

When setting service attributes with application and object message attributes, enter the operation (set or remove) in the `Application` field, and the parameters (service attribute name and value) in the `Object` field.

Table 2-1 Syntax for Custom Message Attributes

| | | |
|---------------------------------------|---------------------------|------------------------------|
| Custom Message Attribute Name | OV_OPERATION ^a | OV_PARAMS ^b |
| Custom Message Attribute Value | SVC_ATTR_SET | [name=value] [name=value]... |
| | SVC_ATTR_REMOVE | [name] [name]... |
| | SVC_ATTR_REMOVE_PREF | [name_prefix] |
| | SVC_ATTR_REMOVE_ALL | |

a. Corresponds to the application message attribute.

b. Corresponds to the object message attribute.

| | | |
|--------|----------------------|--|
| where: | SVC_ATTR_SET | Sets/adds service attributes specified in OV_PARAMS. |
| | SVC_ATTR_REMOVE | Removes service attributes specified in OV_PARAMS. |
| | SVC_ATTR_REMOVE_PREF | Removes service attributes, which name matches the name prefix specified in the OV_PARAMS. |
| | SVC_ATTR_REMOVE_ALL | Removes all service attributes. OV_PARAMS custom message attributes are ignored. |

Message Logging for opccustproc1

Service attributes that are added dynamically are not stored in the database and are therefore lost after the Service Navigator opcservice process has been restarted. Dynamic service-related data can be logged and reused by configuring the opccustproc1 process.

NOTE

This is not possible for service attributes that are added with opcsvcattr.

The format of the logfile is XML. The message ID, the service ID, OV_OPERATION and OV_PARAMS data are stored from each message. The file is stored on the management server at the following location:

```
/var/opt/OV/bin/share/tmp/OpC/mgmt_sv/opccustproc1.xml
```

When the opccustproc1 process is started as part of the ovstart or opcsv -start commands, it reads the messages from the opccustproc1.xml logfile as though they were being received from the message stream interface. Once all stored messages have been processed, opccustproc1 registers with the message stream interface and reverts to normal operation.

To enable logging for the opccustproc1 process on the OVO management server, use the command-line tool ovconfchg. Enter the following:

```
ovconfchg -ovrg <management_server> -ns opc -set \  
OPC_CUSTPROC1_LOG TRUE
```

Where <management_server> is the name of the OVO management server.

The default maximum number of messages that can be written to the opccustproc1 logfile is 1000. You can change this number, to 5000, for example, by using the following command:

```
OPC_CUSTPROC1_LOG TRUE,5000
```

After the maximum number of messages has been reached, for every further message that is appended to the end of the logfile, the earliest message is removed.

Example Template for opcmmsg Message Source

Service Navigator provides the example template `opcmmsg(1|3)STN` for the `opcmmsg(1|3)` message source. Use this example to help you set up a message source template for your message source.

Any `opcmmsg` message matched by this template is fed into the server message stream interface and sent to the management server where the `opccustproc1` process retrieves the message and adds, changes or removes the service attributes accordingly on the Service Navigator GUI.

To Install the Example Template `opcmmsg(1|3)STN`

Perform the following steps to install the example template `opcmmsg(1|3)STN`:

1. Untar the upload tree that contains the example template:

```
tar -xvf /opt/OV/OpC/examples/services/ \
new_stn_upload.tar
```

The upload tree will be placed in the `/tmp/new_stn` directory.

2. Stop the OpenView processes:

```
/opt/OV/bin/ovstop
```

3. Upload the example template:

```
/opt/OV/bin/OpC/opccfgupld -add -index new_stn.idx \
/tmp/new_stn
```

4. Start the OpenView processes:

```
/opt/OV/bin/ovstart
```

5. Start the OVO administrator GUI and open the Message Source Templates window.

The template `opcmmsg(1|3)STN` is now available in the `Toplevel` template group.

6. Assign and distribute the template `opcmmsg(1|3)STN` to the managed nodes.

To Submit Messages for opcmsg(1|3)STN

Use the `opcmsg` command line tool to submit messages for this template.

❑ Example Using Custom Message Attributes:

```
opcmsg a=new_label o=my_obj msg_text="message text" \
service_id=my_service -option OV_OPERATION=SVC_ATTR_SET\
-option OV_PARAMS="[ov_label1=Master] [ov_label2=low]"
```

In this example, a message is submitted that adds two labels for the service `my_service`. The first label has the text `Master` and the second label has the text `low`. The application attribute of this message is set to `new_label` to match the attribute that is set in the template condition.

Figure 2-8 Condition `new_label` of the Example Template `opcmsg(1|3)STN`

The screenshot shows a configuration window titled "Condition No. 2". It is divided into several sections:

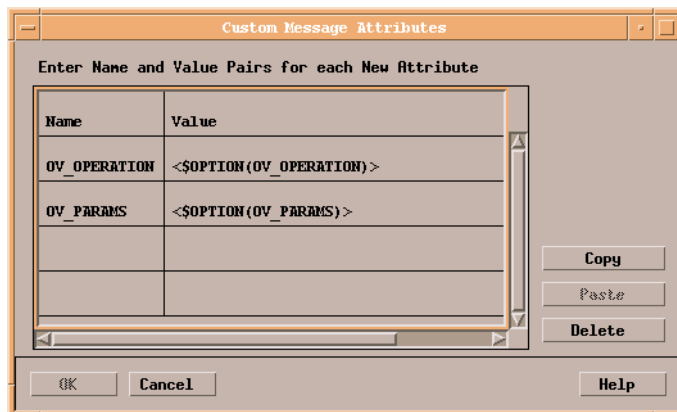
- Description:** Contains the text "Custom Message Attributes".
- Condition:** Contains a table with the following structure:

| Severity | Node | Application | Message Group | Object |
|---|------|-------------|---------------|--------|
| normal warning minor major critical | | new_label | | |

 Below the table is a "Message Text" field.
- Options:** Three radio buttons are present:
 - Suppress Matched Condition
 - = Suppress Unmatched Condition
 - + Message on Matched Condition
- Set Attributes:** Contains a table with the following structure:

| Severity | Node | Application | Message Group | Object |
|-----------------|------|-------------|---------------|--------------------------|
| unchanged | | | | |
| Message Text | | | | |
| Service Name | | | | |
| Message Type | | | | |
| service_changes | | | | Custom Attributes... Ins |

Figure 2-9 Custom Message Attributes Window of the Condition new_label of the Example Template opcmsg(1|3)STN



❑ **Example Using Application and Object Message Attributes:**

```
opcmsg a=SVC_ATTR_SET o="[ov_label1=Master] [ov_label2=low]" \
msg_text="message text" service_id=my_service
```

This example does exactly the same as the example above, but uses the application and object message attributes instead. The application attribute of this message is set to `SVC_ATTR_SET`, which is the operation type for setting the service attributes recognized by the `opccustpro1` and is simultaneously one of the match conditions specified in the `opcmsg(1|3)STN` template.

Figure 2-10 New Condition `opc_msg` with Command and Attributes of the Example Template `opcmsg(1|3)STN`

Description

Application and Object

Condition

| Severity | Node | Application | Message Group | Object |
|---|------|-----------------------|---------------|--------|
| normal warning minor major critical | | SVC_ATTR_SET SVC_ATTR | | |
| Message Text SVC_ATTR_SET SVC_ATTR_REMOVE SVC_ATTR_REMOVE_PREF SVC_ATTR_REMOVE_ALL | | | | |

- Suppress Matched Condition
 = Suppress Unmatched Condition
 + Message on Matched Condition

Set Attributes

| Severity | Node | Application | Message Group | Object |
|---------------------|------|-------------|----------------------|--------|
| unchanged | | | | |
| Message Text | | | | |
| Service Name | | | | |
| Message Type | | | | |
| service_changes | | | Custom Attributes... | Inst |

opccustproc1 in Flexible Management Environments

If you are operating in a flexible management environment, that is with one or more secondary management servers, you need to ensure that the `opccustproc1` process forwards messages from the primary management server to other management servers as defined in the flexible management configuration file `allnodes` that is located on the primary management server.

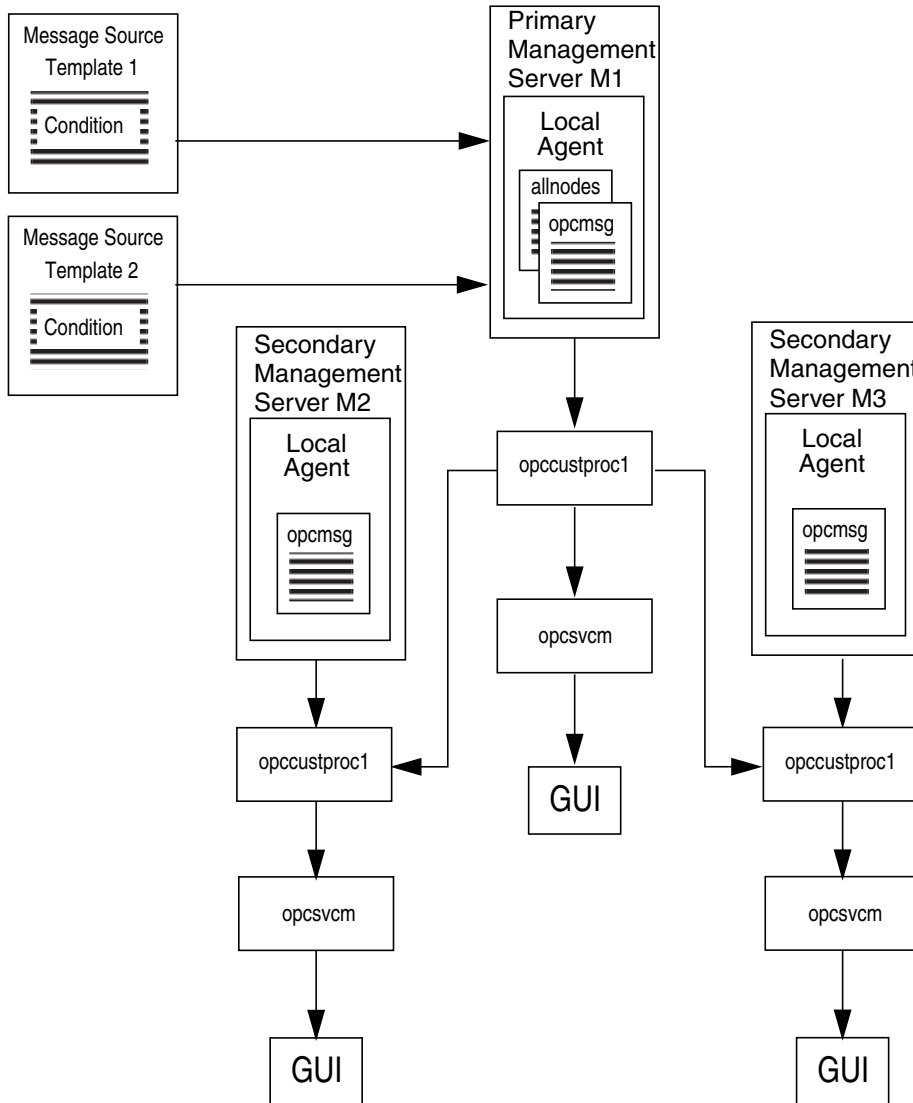
NOTE

The `opccustproc1` and `libopccustproc1.sl` must be running on the primary and all secondary management servers.

The example template `opcmsg(1|3)STN` can be used for this purpose. It contains a condition, `MoM_forward`, that instructs the local agent of the primary management server to send matched OVO messages to the secondary management server.

Figure 2-11 on page 91 shows how `opccustproc1` on the primary management server *M1* forwards messages to the secondary management servers *M2* and *M3*. All management servers have `opcmsg` files installed, while only the primary server has an `allnodes` file installed. It is *not* necessary to have `allnodes` file installed on the secondary management servers for this purpose.

Figure 2-11 Message Flow with `opccustproc1` in Flexible Management Environments



To configure flexible management with `opccustproc1`:

1. Install the example template `opcmsg(1|3)STN` as described on page 86.
2. Make sure that the managed nodes of your primary management server are added to the `Node Bank` and to the `Node Group Bank` of your secondary management server.
3. On the OVO management server, use the command-line tool `ovconfchg`. Enter the following:

```
ovconfchg -ovrg <management_server> -ns opc -set \  
OPC_CUSTPROC1_MOM_FORWARD TRUE
```

Where `<management_server>` is the name of the OVO management server.

4. Restart the OpenView processes:

```
/opt/OV/bin/ovstart
```

5. Copy the flexible management configuration file to the distribution directory:

```
cp /etc/opt/OV/share/conf/OpC/mgmt_sv/tmpl_respmgrs/  
allnodes /etc/opt/OV/share/conf/OpC/mgmt_sv/respmgrs
```

6. Edit the flexible management configuration file `/etc/opt/OV/share/conf/OpC/mgmt_sv/respmgrs/allnodes` and replace the name of the example secondary management server (`tcbbn040.bbn.hp.com`) with the name of your secondary management server.
7. Distribute the template `opcmsg(1|3)STN` to the managed nodes to start a distribution of the `allnodes` file.
8. Submit a test message with `opcmsg` as described in “To Submit Messages for `opcmsg(1|3)STN`” on page 87 to test whether the label changes on both the primary and the secondary management server.

How Do I Start Service Navigator?

Starting the Service Navigator GUI depends on the platform where you want to run the GUI.

If you are running the GUI on a UNIX system, enter:

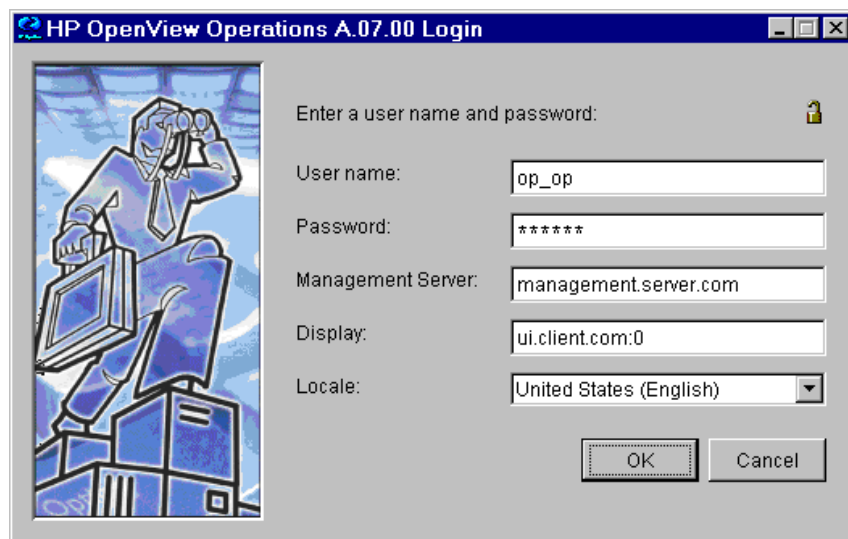
```
/opt/OV/www/htdocs/ito_op/ito_op
```

For more information about the command `ito_op`, see the man page `ito_op(1M)`.

If you are running the GUI on a PC, execute the file `ito_op.bat`.

The OVO Java-based GUI starts and displays a login screen, see Figure 2-12. Enter your user name and password, and the OVO management server you are connecting to. You can also select the language you want to use. This opens the main window of the OVO Java-based GUI.

Figure 2-12 Logging into Service Navigator



How Do I Upgrade Service Navigator?

To upgrade Service Navigator from version A.07.xx to A.08.10, perform the following steps:

1. If service status logging is enabled and you want to migrate service status logs, do this:

- a. Download all service status logs by entering:

```
/opt/OV/bin/OpC/opcsvcdwn -file /tmp/svclogs
```

- b. Store the download data on a tape or separate system.

2. Store the service and service logs repository files on a tape or separate system:

```
/etc/opt/OV/share/conf/OpC/mgmt_sv/opcsvcm/services
```

```
/etc/opt/OV/share/conf/OpC/mgmt_sv/opcsvcm/loggings
```

3. Upgrade the OVO management server to version A.08.10 as described in Chapter 7 of the *OVO Installation Guide for the Management Server*.

NOTE

Service Navigator is installed automatically during the installation of OVO A.08.10 management server.

4. If you have downloaded service status logs, upload them now by entering:

```
/opt/OV/bin/OpC/opcsvcupl /tmp/svclogs
```

5. Restore the service and service logs repository files to their original location on the management server:

```
/etc/opt/OV/share/conf/OpC/mgmt_sv/opcsvcm/services
```

```
/etc/opt/OV/share/conf/OpC/mgmt_sv/opcsvcm/loggings
```

6. Check the configuration and service assignments, enter:

```
/opt/OV/bin/OpC/opcservice -list
```

```
/opt/OV/bin/OpC/opcservice -operators
```

7. Upgrade the OVO Java-based operator GUI to version A.08.10, if it is not already done.

How Do I De-install Service Navigator?

If you no longer require Service Navigator, you can easily de-install it from the GUI client or the management server.

To De-install the Service Navigator GUI Client

NOTE

This procedure de-installs the entire OVO Java-based GUI. If you want to remove *only* services, de-assign them from the operator. Alternatively, you can use `opcservice -remove <service_name>` to remove only one service including subservices, or `opcservice -remove` to remove all services.

❑ PC client

1. Close all running GUIs on the client.
2. Select Start: Settings -> Control Panel. The Windows Control Panel opens.
3. In the Windows Control Panel, double-click the Add/Remove Programs icon. The Add/Remove Programs Properties dialog opens.
4. In the Add/Remove Programs Properties dialog, select HP OpenView for UNIX Java Console and click [Add/Remove ...].

❑ **HP-UX client**

1. Close all running OVO GUIs on the client.
2. De-install Service Navigator interactively, using the `swremove` command. Enter the following:

```
/usr/sbin/swremove OVOPC-SVC
```

3. Check the following logfiles for problems occurring during the de-installation:

- `/var/adm/sw/swagent.log`
- `/var/adm/sw/swremove.log`

❑ **Other UNIX-based systems (for example AIX systems)**

1. Close all running OVO GUIs on the client.
2. Remove the directory `ito_op/` and its contents.

To De-install Service Navigator from an OVO Management Server

To de-install Service Navigator product from the OVO management server:

1. Ensure that all client GUIs accessing Service Navigator are closed.
2. De-install Service Navigator interactively, using the `swremove` command. Enter the following:

```
/usr/sbin/swremove OVOPC-SVC
```

3. Check the following logfiles for problems occurring during the de-installation:

- `/var/adm/sw/swagent.log`
- `/var/adm/sw/swremove.log`

Getting Started

How Do I De-install Service Navigator?

3 The Service Configuration File

In This Chapter

This chapter helps you writing the service configuration file. It contains examples of how the functionality available for Service Navigator can be translated into the configuration syntax. The configuration task can be split into the following steps:

- ❑ Creating the Service Hierarchy
- ❑ Defining the Rules
- ❑ Setting up Service Actions
- ❑ Setting Service Attributes

Creating the Service Hierarchy

Before you start writing the service configuration file you should have a good plan of your service hierarchy and of the relationships between the services. See “Planning Your Service Hierarchy” on page 62 for a list of questions you should consider.

After your hierarchy is drafted, write the configuration file. Begin with setting up the hierarchy and defining the attributes of each service. For example, the icon that represents the service in the GUI, the label that is displayed in the scoping pane or below the icon in the service graph, the title of the service graph window or of the submap displayed in the content area, and so on.

The following example builds the service hierarchy for the SAP/R3 service from the toplevel SAP/R3 service to the ABAP subservice. The service Application Server 01 *contains* its subservices, and the service Application Server 02 *uses* the OS subservice contained in Application Server 01. The name of this relationship is to be displayed in the GUI as *use*, so the XML tag <Label> is used.

The configuration file is available as an example file in the following directory on the management server:

```
/opt/OV/OpC/examples/services/sap.xml
```

See the following tables for more information about the service tags:

- ❑ Table B-3, “The Root Service Tags,” on page 152
- ❑ Table B-4, “The Service Tags,” on page 153
- ❑ Table B-5, “The Source Tags,” on page 155
- ❑ Table B-6, “The Association Tags,” on page 156

The Service Configuration File

Creating the Service Hierarchy

```
<?xml version='1.0' ?>
<!DOCTYPE svcengine SYSTEM "service.dtd">

<Services>
  <Service>
    <Name>SAP</Name>
    <Label>SAP R/3</Label>
    <Title>SAP R/3 Service</Title>
    <Source>
      <Composition/>
      <ServiceRef>application_server</ServiceRef>
    </Source>
  </Service>
  <Service>
    <Name>application_server</Name>
    <Label>SAP Application Server Class</Label>
    <Title>SAP Application Server Class</Title>
    <Icon>/opt/OV/www/htdocs/ito_op/images/server.32.gif</Icon>
    <Source>
      <Composition/>
      <ServiceRef>application_server_01</ServiceRef>
    </Source>
  </Service>
  <Service>
    <Name>application_server_01</Name>
    <Label>Application Server 01</Label>
    <Description>SAP/R3 Application Server01</Description>
    <Source>
      <Composition/>
      <ServiceRef>ABAP</ServiceRef>
    </Source>
    <Source>
      <Composition/>
      <ServiceRef>buffer_space</ServiceRef>
    </Source>
    <Source>
      <Composition/>
      <ServiceRef>processes</ServiceRef>
    </Source>
    <Source>
      <Composition/>
      <ServiceRef>display_queue</ServiceRef>
    </Source>
    <Source>
      <Composition/>
      <ServiceRef>OS</ServiceRef>
    </Source>
  </Service>
  <Service>
    <Name>application_server_02</Name>
    <Label>Application Server 02</Label>
    <Description>SAP/R3 Application Server 02</Description>
    <Source>
      <Dependency/>
      <ServiceRef>OS</ServiceRef>
      <Label>use</Label>
    </Source>
  </Service>
</Services>
```

The previous example uses the `<Source>` tag to establish the service hierarchy. You can also use the `<Association>` tag to achieve a similar effect. The choice depends on your XML coding style. The `<Source>` tag produces nested XML. The `<Association>` tag produces linked XML.

IMPORTANT

Make sure that all services to which you refer in your `<Association>` tags are specified in the same XML file.

❑ **Example using the `<Source>` tag:**

```
<Service>
  <Name>application_server_01</Name>
  <Source>
    <Composition />
    <ServiceRef>ABAP</ServiceRef>
  </Source>
</Service>
```

```
<Service>
  <Name>ABAP</Name>
</Service>
```

❑ **Example using the `<Association>` tag:**

```
<Service>
  <Name>application_server_01</Name>
</Service>

<Service>
  <Name>ABAP</Name>
</Service>

<Association>
  <Composition />
  <SourceRef>ABAP</SourceRef>
  <TargetRef>application_server_01</TargetRef>
</Association>
```

Defining the Rules

After you have the structure of your hierarchy in place, you can think about the status propagation and status calculation rules, as well as any weighting factors you may want to apply. See the section on “What is the Status of a Service?” on page 29 for more information about the general concepts behind status propagation and calculation, and about weighting.

The following example is similar to the previous one. Note, however, that some subservices have been omitted to keep the structure simple and clear. Instead, the rules by which Service Navigator calculates the severity statuses have been added.

Note that the rules have been configured as shared rules. This means that they are defined once and used wherever required. This saves you time and ensures consistency; when you want to change a rule, rather than changing a rule every time it is used, you only need to change the definition once.

❑ **Status propagation rules**

Status propagation rules for the subservices ABAP and OS are defined as increase. In addition, OS is weighted by the factor 2, and a message factor of 2 has been defined for the Application Server 01.

❑ **Status calculation rules**

The status calculation rule for the service Application Server 01 defines multiple thresholds.

See also Table B-7, “The Calculation Rule Tags,” on page 158 and Table B-8, “The Propagation Rule Tags,” on page 160 for a detailed list of the calculation and propagation rule tags.

The configuration file is available as an example file in the following directory on the management server:

```
/opt/OV/OpC/examples/services/sap.xml
```



```

<?xml version='1.0' ?>
<!DOCTYPE svcengine SYSTEM "service.dtd">
<Services>
  <Service>
    <Name>SAP</Name>
    <Label>SAP R/3</Label>
    <Title>SAP R/3 Service</Title>
    <Source>
      <Composition/>
      <ServiceRef>application_server</ServiceRef>
    </Source>
  </Service>

  <Service>
    <Name>application_server</Name>
    <Label>SAP Application Server Class</Label>
    <Title>SAP Application Server Class</Title>
    <Icon>/opt/OV/www/htdocs/ito_op/images/server.32.gif</Icon>
    <Source>
      <Composition/>
      <ServiceRef>application_server_01</ServiceRef>
    </Source>
  </Service>

  <Service>
    <Name>application_server_01</Name>
    <Label>Application Server 01</Label>
    <Description>SAP/R3 Application Server 01</Description>
    <MsgWeight>2</MsgWeight>
    <CalcRuleRef>multiple_thresholds</CalcRuleRef>
    <Source>
      <Composition/>
      <ServiceRef>ABAP</ServiceRef>
      <PropRuleRef>propagation_rule</PropRuleRef>
    </Source>
    <Source>
      <Composition/>
      <ServiceRef>OS</ServiceRef>
      <PropRuleRef>propagation_rule</PropRuleRef>
    </Source>
  </Service>

  <Service>
    <Name>application_server_02</Name>
    <Label>Application Server 02</Label>
    <Description>SAP/R3 Application Server 02</Description>
    <Source>
      <Dependency/>
      <ServiceRef>OS</ServiceRef>
      <Label>use</Label>
      <Weight>2</Weight>
    </Source>
  </Service>

```

The Service Configuration File

Defining the Rules

```
<CalcRule>
  <Name>multiple_thresholds</Name>
  <CalcMultiThreshold>
    <Relative/>
    <Calc>
      <Critical/>
      <Threshold>0.15</Threshold>
    </Calc>
    <Calc>
      <Major/>
      <Threshold>0.5</Threshold>
    </Calc>
    <Calc>
      <Minor/>
      <Threshold>0.6</Threshold>
    </Calc>
    <Calc>
      <Warning/>
      <Threshold>0.7</Threshold>
    </Calc>
  </CalcMultiThreshold>
</CalcRule>
```

```
<PropRule>
  <Name>propagation_rule</Name>
  <Prop>
    <Increase>1</Increase>
  </Prop>
</PropRule>
```

```
</Services>
```

Setting up Service Actions

The previous examples show you how to configure Service Navigator to display parts of the SAP/R3 service hierarchy example used in this manual. To allow operators to fully use Service Navigator, you can also set up service-specific actions which are executed on the node or application which generated the message.

The attributes of a service action include the label which is displayed in the GUI, and the URL or command to be executed. These attributes can be further modified by parameters such as a description, server, service, node, or user. See “The Service Configuration File Syntax” on page 143 for details.

To make these parameters as generic as possible, use OVO variables in the command string when defining service actions. Consider the following example:

You want to define an action with the label `Ping Node` which executes the command `/etc/ping` on the system where your service is running. Enter the following in the service configuration file:

```
<?xml version='1.0' ?>
<!DOCTYPE Services SYSTEM "service.dtd">
<Services>
  <Action>
    <Name>ping</Name>
    <Label>Ping Node</Label>
    <Description>Ping IP addresses</Description>
    <Program>
      <Command>/etc/ping $OPC_SERVICE_VALUE[ip]
      </Command>
    </Program>
  </Action>
</Services>
```

See also Table B-9, “The Service Action Tags,” on page 162 for a detailed list of the service action tags.

Using Attributes in Service Actions

How do you know the IP addresses of the systems where your services are running? Specify them with the tag `<Attribute>` when defining the service in the service configuration file:

```
<?xml version='1.0' ?>
<!DOCTYPE Services SYSTEM "service.dtd">
<Services>
<Service>
  <Name>application_server</Name>
  <Label>SAP Application Server Class</Label>
  <Title>SAP Application Server Class</Title>
  <Icon>/opt/OV/www/htdocs/ito_op/images/server.32.gif
</Icon>
  <Attribute>
    <Name>ip</Name>
    <Value>70.154.198.255</Value>
  </Attribute>
  <ActionRef>ping</ActionRef>
  <Source>
    <Composition/>
    <ServiceRef>application_server_01</ServiceRef>
  </Source>
</Service>
</Services>
```

Setting up Local Actions

Service actions can also be executed locally on the client where the Service Navigator GUI is running. For example, to find out the hostname of the client where the Service Navigator GUI is currently running, set up the following action. This executes the command `hostname` on the Service Navigator GUI client.

```
<?xml version='1.0' ?>
<!DOCTYPE Services SYSTEM "service.dtd">
<Services>
  <Action>
    <Name>Hostname</Name>
    <Label>Local Action</Label>
    <Description>Get hostnames</Description>
    <Program>
      <Command>hostname</Command>
      <Localnode/>
    </Program>
  </Action>
</Services>
```

Setting Service Attributes

Service attributes are characteristics of a service that can be defined in the service configuration file. They are used to carry additional information about a service.

Service Navigator can also handle attributes dynamically. These are attributes that are added at runtime and can override the values of attributes of the same name defined in the service configuration file. Updating attribute values at runtime can create dynamic effects in the Service Navigator GUI, for example changing labels of a service icon.

The following XML file shows you how you can change the appearance of a service submap with service attributes in a service configuration file. See “... How to Enhance a Service Submap?” on page 128 for details.

The XML shown below is extracted from the example configuration file `/opt/OV/OpC/examples/services/italy_geo.xml`. In this example, the attributes of the service `Italy Banana HQ` define a map of Italy as service submap background. This map has a width of 722 pixels and a height of 792 pixels. The subservice `Friuli branch` is positioned at 385 by 85 pixels on this background.

```
<?xml version="1.0"?>
<Services xmlns="http://www.hp.com/OV/opcsvc">
  <Service>
    <Name>italy_geo</Name>
    <Label>Italy Banana HQ</Label>
    <Icon>/opt/OV/www/htdocs/ito_op/images/eye.gif</Icon>
    <Attribute>
      <Name>ov_background</Name>
      <Value>italy.jpg</Value>
    </Attribute>
    <Attribute>
      <Name>ov_map_width</Name>
      <Value>722</Value>
    </Attribute>
    <Attribute>
      <Name>ov_map_height</Name>
      <Value>792</Value>
    </Attribute>
    <Source>
      <Composition/>
      <ServiceRef>Friuli branch</ServiceRef>
    </Source>
  </Service>
</Services>
```

The Service Configuration File

Setting Service Attributes

```
        </Source>
</Service>
<Service>
  <Name>Friuli branch</Name>
  <Label>Friuli branch</Label>
  <Icon>/opt/OV/www/htdocs/ito_op/images/banana.gif</Icon>
  <Attribute>
    <Name>ov_posX</Name>
    <Value>385</Value>
  </Attribute>
  <Attribute>
    <Name>ov_posY</Name>
    <Value>85</Value>
  </Attribute>
</Service>
```

When you have set your service attributes in the XML configuration file, use the command `opcservice` to load the new configuration. Refer to “The `opcservice` Command” on page 168 and the `opcservice(1M)` man page for more information.

Setting up Service Assignments

The last step in configuring Service Navigator is the operator-to-service assignment. If you do not want to do this with the `opcservice` command, you can define the assignments in the service configuration file itself. Note, however, that using `opcservice` is the recommended way.

The following example assigns the service `SAP` to the operator `ito_op`. The advantage of this method is that you can customize the Service Navigator GUI for each operator by defining further attributes of the services for which the operator is responsible. See also Table B-10, “The Operator Tags,” on page 164 for a detailed list of the operator tags.

The configuration file is available as an example file in the following directory on the management server:

```
/opt/OV/OpC/examples/services/sap.xml
```

The Service Configuration File

Setting up Service Assignments

```
<?xml version='1.0' ?>
<?xml version='1.0' ?>
<!DOCTYPE svcengine SYSTEM "service.dtd">
<Services>
  <Service>
    <Name>SAP</Name>
    <Label>SAP R/3</Label>
    <Title>SAP R/3 Service</Title>
    <Source>
      <Composition/>
      <ServiceRef>application_server</ServiceRef>
    </Source>
  </Service>

  <Service>
    <Name>application_server</Name>
    <Label>SAP Application Server Class</Label>
    <Title>SAP Application Server Class</Title>
    <Icon>/opt/OV/www/htdocs/ito_op/images/server.32.gif</Icon>
    <Source>
      <Composition/>
      <ServiceRef>application_server_01</ServiceRef>
    </Source>
  </Service>

  <Service>
    <Name>application_server_01</Name>
    <Label>Application Server 01</Label>
    <Description>SAP/R3 Application Server 01</Description>
    <MsgWeight>2</MsgWeight>
    <CalcRuleRef>multiple_thresholds</CalcRuleRef>
    <Source>
      <Composition/>
      <ServiceRef>ABAP</ServiceRef>
      <PropRuleRef>propagation_rule</PropRuleRef>
    </Source>
    <Source>
      <Composition/>
      <ServiceRef>OS</ServiceRef>
      <PropRuleRef>propagation_rule</PropRuleRef>
    </Source>
  </Service>

  <Operator>
    <Name>ito_op</Name>
    <Label>ito_op's services</Label>
    <Description>Services of operator ito_op</Description>
    <ServiceRef>SAP</ServiceRef>
  </Operator>
```

A **Did You Know...**

In This Chapter

This section contains tips and tricks for configuring Service Navigator:

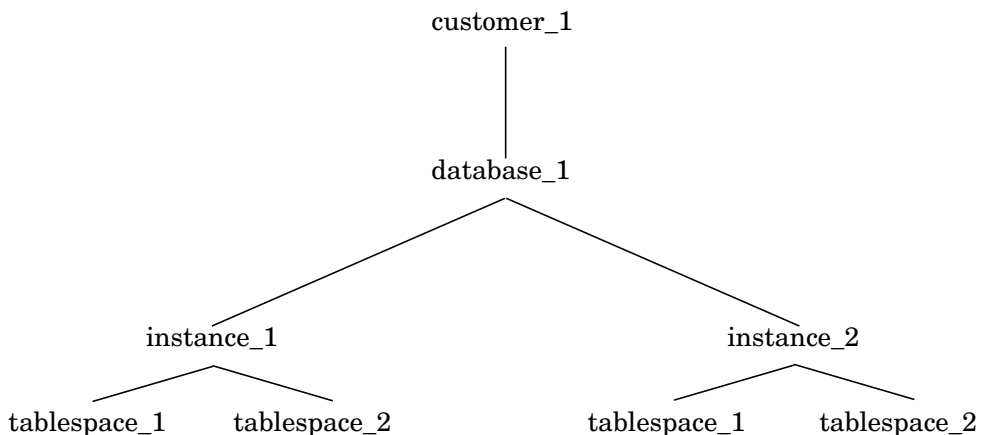
- ... That Names are Important?
- ... How to Use OVO Variables?
- ... That You Can Use Example Configuration Files?
- ... That You Can Customize Icons and Backgrounds?
- ... How to Label Service Icons?
- ... How to Enhance a Service Submap?
- ... That You Can Operate with Service Navigator GUI From Other Applications?
- ... That You Can Personalize Your Service Graph and Service Map View?

... That Names are Important?

Service names are unique identifiers or strings which you can choose freely when you write the configuration file. They do become important when you enter them in the OVO administrator GUI to tell OVO which messages match which services. Note, however, that you do not have to create a new template or condition for each service you are monitoring. If you devise a structured naming schema for your services, you can use OVO's predefined variables to match the service name attached to the message with the service names set for Service Navigator. This has the advantage that you can keep your OVO templates generic.

Consider the following example: your IT company is managing several database installations for different customers. You know that each database installation can have several instances, and that each instance has several tablespaces which you want to monitor. You also know that you want to use the same OVO template to monitor the tablespaces. So you need to come up with a naming schema that enables you to create unique service names, *and* enables you to match these names with your template. Your service hierarchy draft would look similar to the one in Figure A-1:

Figure A-1 **A Service Hierarchy Draft**



When you know the general layout you can think about the names. Think of what makes each name unique and then compose the name with this information. In the example in Figure A-1 on page 115, the customer name and the system name on which the database is installed would uniquely identify your service.

How can you tell OVO without hardcoding the names in the GUI? OVO can find out some of this information by resolving variables. For example, you can use the OVO variable `$MSG_NODE_NAME` which would return the name of the node to which the template is assigned.

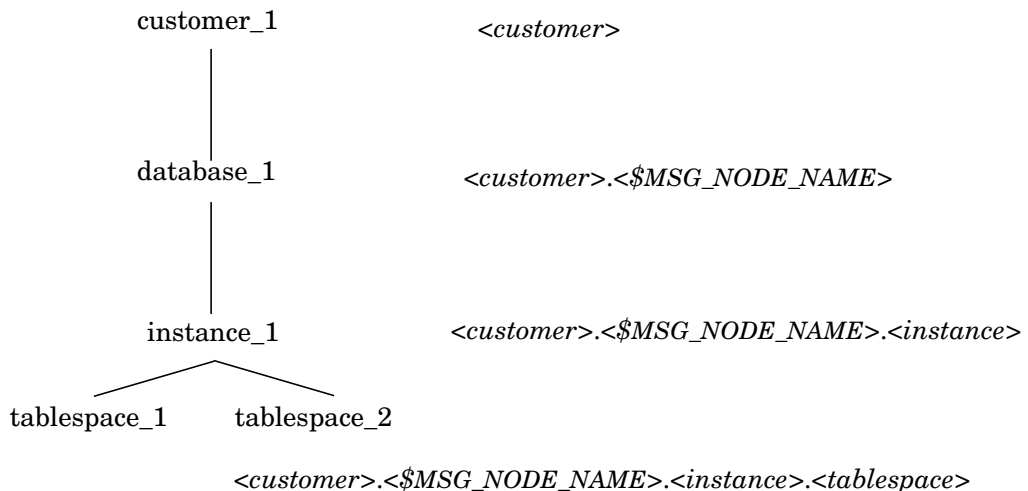
How can you find out the customer name? This depends on the type of template you are using. If you were using a logfile template and the name of your customer was included in the logfile, you could use OVO pattern-matching mechanism to retrieve the customer name. You would enter `<*.customer>` in the Message Text field of the Condition No. window, and would enter `<customer>` in the Service Name field. OVO then matches `customer` with the name of customer and would enter it in the Service Name field.

In general, you can match any text that your template or condition provides.

Figure A-2 shows the naming schema for your database monitoring services:

Figure A-2

A Naming Schema



To receive messages for the service `tablespace_1`, enter the following in the `Service Name` field in the OVO GUI:

`<customer>.<MSG_NODE_NAME>.<instance>.<tablespace>`

And define its name in the service configuration file as:

`Company.system01.your.tech.com.instance_1.tablespace_1`

... How to Use OVO Variables?

OVO allows you to use variables as part of the service name in the Service Name field.

For example, consider the following services:

```
SAP:applsv01
```

```
SAP:applsv02
```

To intercept messages from all application servers, enter `SAP: <$MSG_NODE_NAME>` in the Service Name field.

The message condition resolves the variable to `applsv01` or `applsv02` depending on where the message originated, and sends it to the corresponding service. See also Figure 2-3 on page 70 for an example.

See the *OVO Administrator's Reference* for a list of variables that can be used in this field. Note that you cannot use these variables in the service configuration file.

... That You Can Use Example Configuration Files?

Service Navigator comes with a selection of service configuration files which you can copy and edit at your leisure. The example files show how to use the configuration syntax and how to apply Service Navigator to your environment. The examples are available in the following directory on the OVO management server:

```
/opt/OV/OpC/examples/services/
```

This directory also contains a README file that lists all available examples and their related files. The following examples are available:

Table A-1

Example Configuration File

| File Name | Description |
|----------------------------|---|
| action.xml | Example of how to use actions. |
| banking.xml ^a | Banking services example. |
| carsupply.xml ^a | Supply chain services example of the car industry. |
| cltsvr.xml | Client-server system with database and nodes. |
| cluster.xml ^a | MC/ServiceGuard cluster system. |
| diskless.xml ^a | NFS diskless sample service. |
| email.xml ^a | Email sample service using resources. |
| factor.xml | Factor usage sample service. |
| fileserv.xml | File server example with file systems and daemons. |
| georga.xml | Geographically and organizationally structured servers. |
| icons.xml | Service file that shows all available custom icons. |

Table A-1 Example Configuration File (Continued)

| File Name | Description |
|----------------------------|--|
| inet.xml | Network services. |
| isp.xml | Internet Service Provider service with customers. |
| local.xml | Example of how to use local service actions. |
| lvm.xml ^a | Logical Volume Manager example service. |
| oracle.xml ^a | Oracle tablespace monitoring example. |
| outage.xml ^a | Outages from Service Navigator as service actions. |
| perf.xml | Integrating performance applications and messages. |
| redundant.xml ^a | 4 redundant servers where at least 3 must be running |
| res.xml | Shared and private resources. |
| sap.xml | The SAP/R3 example from this manual. |
| vpo.xml ^a | HP OpenView Operations as service. |

- a. This configuration file can be generated automatically with the shell scripts available in the same directory.

If element values for service files are not specified, warning messages are displayed. To handle error messages, see “Service Files Error Checking” on page 67 for more details.

The file `svcapps.tar` is an tar archive of an OVO upload tree that can be uploaded into OVO. It installs an application group with applications that list, assign, deassign, and remove services and assignments. Add, check and replace operations are not included because they require a file as input parameter which is not accessible as a parameter in the OVO administrator GUI.

1. Change directory:

```
cd /var/opt/OV/share/tmp/OpC_appl/
```

2. Untar the file:

```
tar -xvf /opt/OV/OpC/examples/services/svcapps.tar
```

3. Upload the upload data:

```
opccfgupld -add /var/opt/OV/share/tmp/OpC_appl/svcapps
```

... That You Can Customize Icons and Backgrounds?

You can customize the Service Navigator GUI by changing the default background, tree icons, and icons in the service graph. This is done by specifying your choice in the service configuration file, see “The Service Configuration File Syntax” on page 143.



Note that some example image files are at your disposal in the following directory on the management server after the Service Navigator installation:

```
/opt/OV/www/htdocs/ito_op/images/
```

All available icons are used in the example service configuration file `/opt/OV/OpC/examples/services/icons.xml`; activate this file to display each icon in the Service Navigator GUI.

Table A-2 gives an overview of the areas you can customize.

Table A-2 Customizing the Service Navigator GUI

| Area | Tag | Input | Size in Pixels | Type | Default |
|--------------------|--------------|--------------------------------------|--|------------------|---|
| Background graphic | <Background> | URL or path on the management server | any; graphics are resized to fit the GUI | GIF | white |
| Tree icon | <Icon> | URL or path on the management server | 32 x 32; resized to 16 x 16 | GIF ^a |  |
| Graph icon | <Icon> | URL or path on the management server | 32 x 32; resized to 16 x 16 | GIF ^a |  |

- a. Your icons must have a transparent background to properly display the severity status.

... That You Can Customize Icons and Backgrounds?

The URLs for these icons and graphics must be well-formed. You can choose between the following transport modes:

To retrieve files through a web server:

```
http://<$OPC_MGMTSV>:3443/ITO_OP/images/<icon.gif>
```

To retrieve files from the management server file system:

```
/opt/OV/www/htdocs/ito_op/images/<icon.gif>
```

You can either specify an absolute or relative path name.

Note that you can use variables as part of the URL. See “OVO Variables in URL Definitions” on page 175 for more information.

... How to Label Service Icons?

Labels can be added to service icons to display additional information about the service. A label can be text or an image in service presentations and can apply to all service information, such as graphs, custom service maps, service submaps, impacted services and root causes, and service icons in the shortcut bar.

Services can be labeled in two ways using reserved service attributes that are recognized by the GUI:

❑ **Service Configuration File**

Specify labels in the service configuration file. The values for these labels are acquired at Service Navigator start-up and cannot be changed or removed during the current session. See “Setting Service Attributes” on page 109 for an explanation of specifying labels using the service configuration file. Labels displayed from service configuration data can be overridden dynamically.

❑ **Dynamically**

It is also possible to continuously update the text or image of a label and display the most up-to-date information by the use of messages. The refresh rate is determined by that set for the message browser. The methods of automatically setting and updating are detailed in “Setting Service Attributes Dynamically” on page 75.

The refresh interval of the GUI is 5 seconds or longer. All visible changes in the GUI are triggered by the refresh cycle, dynamic changes cannot be faster than 5 seconds from initiation.

The information represented in dynamic labels is usually retrieved from messages. The text or an image file name, parsed from, for example, message text, can then be used in a service label in the Service Navigator GUI.

Syntax for Dynamic Labels

Dynamic labels use the concept of dynamically set service attributes to achieve an effect where the content and position of labels displayed for service icons changes dynamically. For this purpose, Service Navigator uses predefined reserved service attribute names. The following table explains this syntax:

Table A-3 Key Attribute Names for Dynamic Labels

| To Change: | Label Text | Label Image |
|------------------|---|---|
| Attribute Name: | <code>ov_label<n></code> | <code>ov_label<n></code> |
| | <n> is a number that determines the position of the label below the service icon. Must be a positive integer. | |
| Attribute Value: | <code><string></code> | <code>ov_image:<path_to_image></code> |
| | <code><string></code> must be a positive integer. <code><path_to_image></code> is the full path to the image in the file system. | |

Text Labels

The service icon labels have a maximum width that is sufficient to display approximately 25 characters. If the label is too long to be fully displayed, it is cut short and appended with three dots (. . .). The tool-tip on the label shows the complete label string.

Image Labels

Image indicators using graphic files instead of text can be used to represent service information in labels. These images are typically gauge graphs as seen in Figure 3-1 on page 126, pie charts, or icons, but can be any kind of image that represents information.

Valid image file formats are gif or jpeg.

If an image is to be used as a label, the syntax is of the form:

`ov_image:<path_to_image>`

The file name must be specified with the full path name or with the URL (http or ftp).

It is advisable to store the image files on the OVO management server in the directory:

```
/opt/OV/www/htdocs/ito_op/images/
```

where they can be accessed without the full path specified, or a subdirectory of this directory.

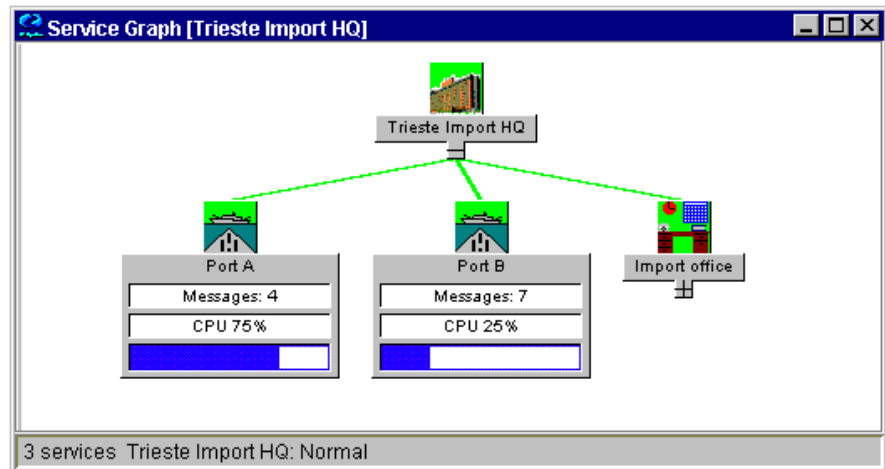
It is also advisable to use logical file names to make later application easier. For example, a set of level gauge files could be called:

```
gauge0.gif, gauge25.gif, gauge50.gif, gauge75.gif and  
gauge100.gif.
```

An Example for Setting Labels Dynamically

Figure 3-1 shows an example of how additional labels can be applied. The following command examples describe how to set up these labels. These have however been entered using `opcsvcat`. To be of greatest value, these should be entered as variables through message attributes so as to update fully automatically.

Figure 3-1 Labeling a Service



The following commands show how these labels were applied. If the command is executed successfully, a check is made and an OK message is displayed.

```
# /opt/OV/bin/OpC/opcsvcatr \  
svc_id="Port A" \  
name=ov_label1 value="Messages: 4" \  
name=ov_label2 value="CPU 75%" \  
name=ov_label3 value=ov_image:\  
/opt/OV/www/htdocs/ito_op/images/gauges/gauge75.gif
```

```
<Results>  
<OK/>  
</Results>
```

```
# /opt/OV/bin/OpC/opcsvcatr \  
svc_id="Port B" \  
name=ov_label1 value="Messages: 7" \  
name=ov_label2 value="CPU 25%" \  
name=ov_label3 value=ov_image:\  
/opt/OV/www/htdocs/ito_op/images/gauges/gauge25.gif
```

```
<Results>  
<OK/>  
</Results>
```

... How to Enhance a Service Submap?

A service submap can be customized so that the service icons are placed on a preselected background and positioned to correspond to their real world location. A submap can be configured individually for each service.

It is advisable to define backgrounds and icon locations in the service configuration file. See “Setting Service Attributes” on page 109 for an explanation of specifying backgrounds and icon positions using the service configuration file. However, it is possible to update these dynamically, as explained in “Setting Service Attributes Dynamically” on page 75.

You can configure the following aspects of a service submap:

❑ **Background image**

A graphic file with one of the following formats: gif or jpeg. For example, you could choose an image representing a geographical map of the country where your service is located.

If a background graphic is not specified, the default service submap is displayed. You cannot change the image specification in the service configuration file through the service navigator GUI.

❑ **Background size**

You define the size of the background image in pixels, usually in the service configuration file. The size is defined by the image width and height and cannot be changed in the service configuration file through the service navigator GUI.

❑ **Subservice position**

You can define the position of subservices on the background image of the parent service. The position is defined by specifying values on the X and Y axes, which applies to the display size of the parent service background image.

Any service can have any or all of the above service attributes (Background, Map Size and Position) specified, as any service can simultaneously be a parent service and a subservice.

Figure A-3 An Enhanced Service Submap

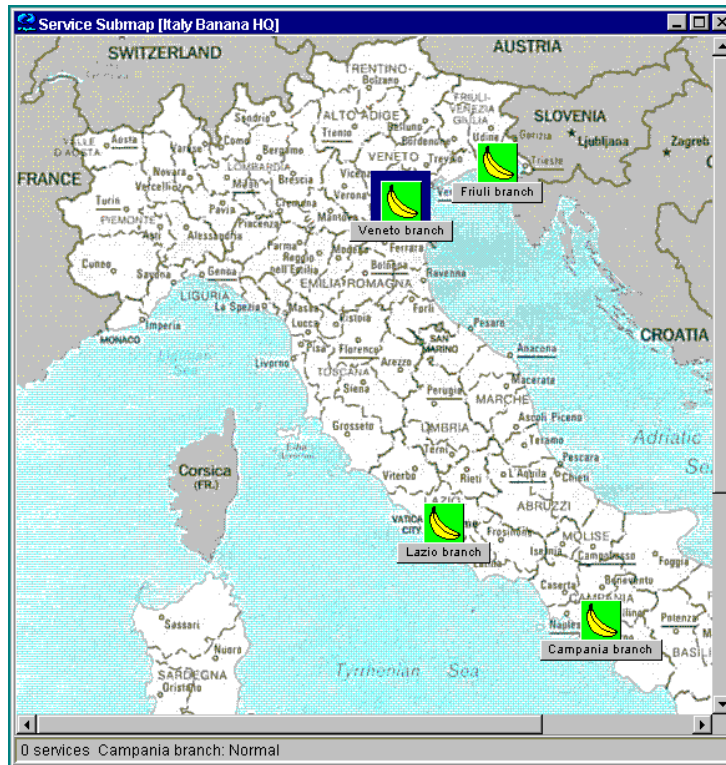


Table A-4 Key Attribute Names for Service Submap Backgrounds

| To Configure | Background | Background Size | Subservice Position on Background |
|-----------------|---|--|--|
| Attribute Name | ov_background | ov_map_width ov_map_height | ov_posX ov_posY |
| Attribute Value | ov_image:<path_to_image> <path_to_image> is the full path to the image in the file system. | <string> <string> must be a positive integer. The value is defined in pixels. | <string> <string> must be a positive integer. The value is defined in pixels. |

... That You Can Operate with Service Navigator GUI From Other Applications?

You can operate with the Service Navigator GUI remotely from other Java applications using the Java GUI Remote APIs. These applications can run in one of the following ways:

- ❑ separately from the Service Navigator GUI
- ❑ as an integrated add-on components in the workspace of the Service Navigator GUI

Java GUI Remote APIs are useful if you have mapped OVO services in other applications because they enable you to resolve problems without searching for problem causing elements directly in the Service Navigator GUI.

For details about the available Remote APIs, refer to the Java GUI Remote APIs Specification, which can be accessed through the following URL:

`http://<management_server>:3443/ITO_OP`

In this instance, `<management_server>` is the fully qualified hostname of your management server.

For more information on the concept, integration details and usage of the Java GUI Remote APIs, refer to *OVO Application Integration Guide*.

When the Java GUI Remote APIs are enabled, an additional icon is displayed in the Service Navigator GUI. Refer to the *OVO Java GUI Operator's Guide* for more information.

... That You Can Personalize Your Service Graph and Service Map View?

When operating with many service graphs and maps, especially large ones, it can become difficult to access a particular service or to easily navigate through the graph or map.

You can personalize your service graph and map views in Service Navigator to obtain a more focused overview of the services and make easier to work with them.

You can achieve this by using the following Service Navigator features:

- Service Graph and Map Zooming
- Navigating through Service Graphs and Maps
- Easy Scrolling and Repositioning of Services in Service Graphs
- Making Selections in Service Graphs
- Customizing Font Size in Service Graphs and Maps

NOTE

Zooming and navigating are supported in service graphs (including Service Root Cause Graph and Service Impact Graph) and in service maps. Easy scrolling, repositioning of services, and making selections are supported *only* in service graphs.

Service Graph and Map Zooming

You can set the size of service graphs and maps using the manual and automatic zooming adjustments of the following zooming commands:

- Service Graph and Map Quick Zooming
- Setting Default Zoom Ratio

Service Graph and Map Quick Zooming

You can instantly magnify or reduce your view of the service graph or map in the following ways:

Using the ‘zoom’ drop-down list from the toolbar

From the ‘zoom’ toolbar drop-down list, select a value from a predefined set of proportional zoom ratios expressed in percentages, from 5% to 100%.

There are two other zoom options available from the ‘zoom’ toolbar drop-down list:

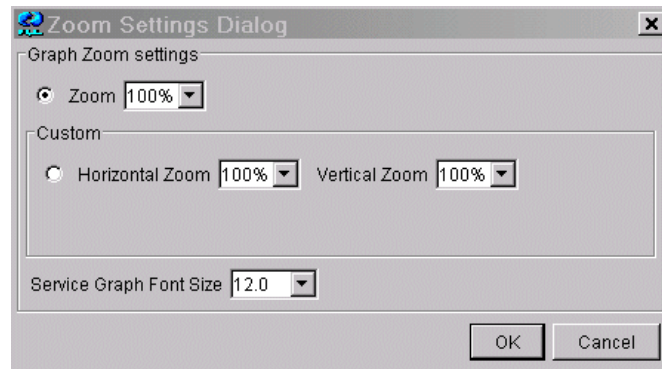
- Fit

Fit zooms the whole service graph or map, so that it fits exactly into the current window.

- Custom

Custom opens the Zoom Settings Dialog window, where you can specify the overall zoom ratio, the horizontal, and the vertical zoom ratio, as shown in the Figure A-4.

Figure A-4 Zoom Settings Dialog Window



Using the Zoom Settings pop-up menu

The options available from the the ‘zoom’ toolbar drop-down list are also available from the Zoom Settings pop-up menu.

To raise the Zoom Settings pop-up menu, right-click any empty (white) area of the Service Graph or Service Submap window.

Setting Default Zoom Ratio

You can set the zoom ratio that is used each time a new service graph or map is opened in the `Services` tab of the Java GUI

Edit-> Preferences dialog.

Refer to *OVO Java GUI Operator's Guide* for a description of each option on the Preferences dialog box.

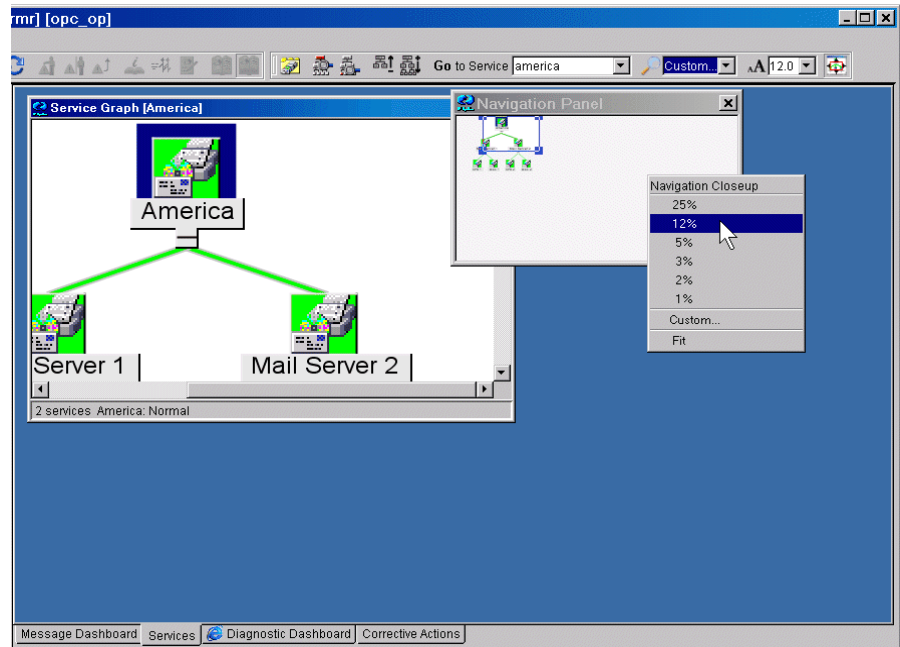
Navigating through Service Graphs and Maps

Navigating Using Navigation Panel

The Navigation Panel is a resizable window which helps you to easily navigate through the whole service graph or map area, and to get a better overview of the services.

Figure A-5 represents the Navigation Panel and Service Graph windows in the Service Navigator workspace.

Figure A-5 Service Graph Navigation Panel



The view box in the Navigation Panel represents the boundaries of the service graph area that is displayed in the Service Graph window.

NOTE

A Navigation Panel in one pane has no influence on another pane. If you want to navigate through service submaps that are split into two horizontal panes, such as service submaps with a geographical display, you must use the Navigation Panel in each pane to navigate through that pane.

When two service (sub)maps are displayed, you can switch the target for the Navigation Panel by clicking the title bar of a service (sub)map where you want to navigate.

If the entire service graph or map is not visible in the current workspace, you can use the Navigation Panel to bring another area of the service graph into view by dragging the view box over the Navigation Panel.

It is also possible to magnify or reduce the size of the service graph or map area visible in the Service Graph or Service Submap window. You can achieve this by manually resizing the view box in the Navigation Panel.

You can also magnify or reduce the size of the Navigation Panel by using the Navigation Closeup pop-up menu as shown in the Figure A-5.

NOTE

You can also use the Navigation Panel to navigate through Service Impact Graph and Service Root Cause Graph windows.

Navigation Panel can be enabled or disabled and you can do this in one of the following ways:

- By selecting the View-> Navigation Panel from the menu bar.
- By choosing the Toggle Navigation Panel icon from the toolbar.

Navigating from the Toolbar

NOTE

Go to Service drop-down list is available only from the toolbar list.

... That You Can Personalize Your Service Graph and Service Map View?

Go to Service drop-down list is a part of the Service Toolbar component. It contains a list of all services currently present in Java GUI cache that can be displayed within selected service graph.

NOTE

You can use Go to Service drop-down list for navigating only from service graphs.

When you select a service and press the [Go] button on the service toolbar, the service is displayed in the center of a service graph surrounded by its neighbouring services. This way you can easily navigate through your service graph.

NOTE

If the Service Load on Demand option from the Edit-> Preferences dialog is enabled, only the names of the services that are currently retrieved from the management server are displayed in the Go to Service drop-down list.

Making Selections in Service Graphs

In large service graphs, with many service icons and connection lines, it can be difficult to determine relationships between service icons.

To easily distinguish between service icons and determine the relationships among them, you can do the following:

Selecting service icons

When you select a service icon, all connection lines to its subsequent services are highlighted, as shown in the Figure A-6 on page 139.

Multiple selections of service icons, either using the CTRL key or mouse (lasso selections), are also possible.

Highlighting connection lines

It is possible to highlight any connection line among service icons to determine their relationships.

NOTE

You can select the service icons and their connection lines only in active Service Graph windows. When the Service Graph window becomes inactive, all selections are no longer highlighted.

Easy Scrolling and Repositioning of Services in Service Graphs

The following features make your work with the service graphs easier:

❑ **Easy scrolling**

You can scroll the whole service graph area by dragging it with the mouse, without need to use the scroll bars. The dragging continues even if you move the mouse outside the Service Graph window.

❑ **Repositioning of service icons**

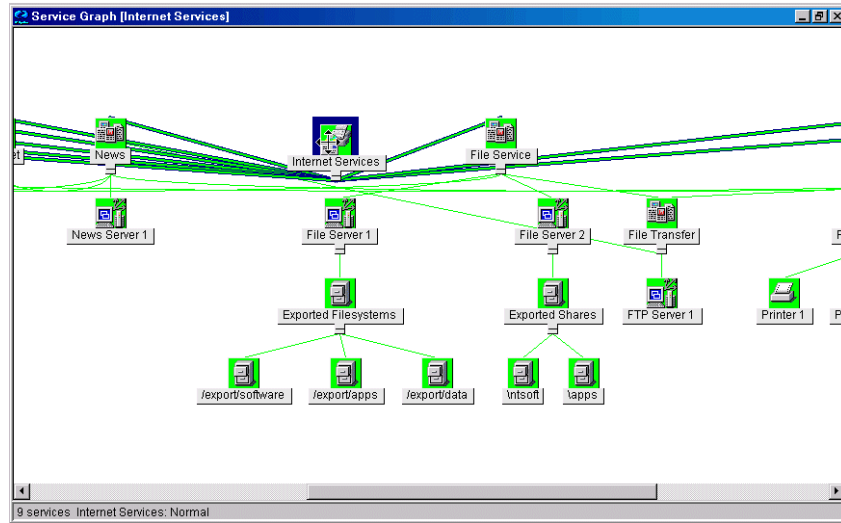
You can reposition service icons and their connection lines by using drag-and-drop. If more than one service icons are selected, you can simultaneously move them over the service graph area.

These features make it much easier to determine the connections that belong to a particular services as well as allow you to personalize the service graph shape. Figure A-6 on page 139 shows a repositioned Internet Services icon.

NOTE

It is possible to move service icons even outside service graphs and maps, as well as it is possible to move service graphs and maps themselves outside the workspace pane. To learn more about Java GUI drag and drop operations, and about the possible ways how they can be performed, refer to the *OVO Java GUI Operator's Guide*.

Figure A-6 **Repositioning Service Icons**



NOTE

Any action that changes the service graph structure, such as expand or collapse, restores the service icons to their default positions.

Customizing Font Size in Service Graphs and Maps

You can customize the font size in service graphs and maps by setting it in the following ways:

❑ **Using the ‘font size’ drop-down list from the toolbar**

You can change the font size in your currently active service graph or map by choosing one of the available values from the ‘font size’ toolbar drop-down list.

❑ **By specifying the value in the Zoom Settings Dialog window**

You can set the font size in your currently active service graph or map, by specifying a value in the Zoom Settings Dialog window.

NOTE

In the Zoom Settings Dialog window, you can also set the overall zoom ratio, the horizontal, and the vertical zoom ratio, as shown in the Figure A-4 on page 132.

The Zoom Settings Dialog window opens when you choose Custom option from the ‘zoom’ toolbar drop-down list, or from the Zoom Settings pop-up menu in your Service Graph or Service Submap window. To learn more about Zoom Settings Dialog window, see “Service Graph and Map Quick Zooming” on page 132.

❑ **By setting the default value in the Preferences dialog**

You can set the font size that is used each time a new service graph or map is opened in the Services tab of the Java GUI Edit-> Preferences dialog.

Refer to *OVO Java GUI Operator’s Guide* for a description of each option on the Preferences dialog box.

NOTE

You can also set the general font size, which applies to all Java GUI components except for the service graphs and maps. Refer to *OVO Java GUI Operator’s Guide* for more information.

B **Reference Information**

In This Chapter

This section contains detailed information about the XML syntax used in the service configuration file. Additionally, all options available for the `opcservice` command are documented.

- ❑ The Service Configuration File Syntax
- ❑ The `opcservice` Command
- ❑ OVO Variables
- ❑ Service Navigator Man Pages
- ❑ Service Navigator Product Bundles

The Service Configuration File Syntax

The service configuration file is originally a file in Extensible Markup Language (XML) format. You can edit the file using any text or XML editor. When you have finished working on this file, use the command `opcservice` to update the OVO configuration with your additions or modifications. `opcservice` reads the service configuration file and moves it into the following directory:

```
/etc/opt/OV/share/conf/OpC/mgmt_sv/opcsvcm/services
```

NOTE

Never edit this file directly.

Defining a service configuration includes the following aspects:

- Defining the Services and Associations
- Defining Status Calculation and Propagation Rules
- Defining Service Actions
- Assigning Services to Operators

Notation Used

The format of the service configuration file is based on the World Wide Web Consortium Extended Markup Language (XML).

The Document Type Definition (DTD) for the Service Navigator XML syntax is printed in this section and is also available on the management server in:

```
/etc/opt/OV/share/conf/OpC/mgmt_sv/dtds/service.dtd
```

This DTD is also available in XML Schema Definition (XSD) format in the same directory. This alternative format is based on XML and therefore easier to read with XML editors.

The following syntax rules apply:

❑ **Case Sensitive**

The Service Navigator XML parser is case sensitive; the XML tags must be specified as defined in the DTD.

❑ **XML Processing Instructions**

Each XML file must *start* with an XML processing instruction:

```
<?xml version='1.0' ?>
```

Comments or any other tag are not allowed before this instruction.

❑ **Codeset**

If no codeset is defined, the default value UTF-8 is used.

```
<?xml version="1.0" encoding="UTF-8"?>
```

Instead of UTF-8, the following codesets can be used with Service Navigator:

Table B-1 Supported Codesets

| Language | Codeset |
|--|----------------|
| Czech | ISO-8859-2 |
| Japanese | Shift_JIS |
| Korean | EUC-KR |
| Russian | ISO-8859-5 |
| Simplified Chinese | GB2312 |
| Traditional Chinese | Big5 |
| Western European (for example, English, French, German, or Spanish) | ISO-8859-15 |
| | UTF-8 |

The codesets are the standard codeset names as defined by IANA (Internet Assigned Numbers Authority).

❑ **Name Spaces**

The following namespace is used by the Service Navigator service DTD: `http://www.hp.com/OV/opcsvc`

Name spaces are specified within the toplevel XML tag and are used to uniquely identify the XML tags. For example, a file `services.xml` should start like this:

```
<?xml version='1.0' ?>
<Services xmlns="http://www.hp.com/OV/opcsvc">
```

- ❑ **Comments**

XML provides a mechanism for commenting code which has the same syntax as for HTML comments: `<!-- comment -->`; comments must not occur within declarations or inside element tags.

- ❑ **Content Model Operators**

The following content model operators occur in the DTD:

Table B-2

XML Content Model Operators

| Symbol | Usage |
|--------|-------------------------|
| , | Strict ordering |
| | Selection |
| + | Repetition; minimum one |
| * | Repetition |
| ? | Optional |
| () | Grouping |

- ❑ **#PCDATA**

Elements that have character content are declared as #PCDATA.

- ❑ **#ASCII**

Elements that have ASCII only character content are declared as #ASCII.

- ❑ **EMPTY**

Elements that are empty are declared as EMPTY.

```
<!-- XML DTD for service engine files -->
<!-- Services is the root element -->

<!ELEMENT Services (CalcRule | PropRule | Action
                    | Service | Operator | Association)* >

<!ENTITY % Severity "(Normal | Warning | Minor | Major | Critical)" >
<!-- Status Calculation Rules -->
<!ELEMENT CalcRule (Name,
                    (CalcMostCritical |
                     CalcSingleThreshold |
                     CalcMultiThreshold)) >
<!ELEMENT CalcMostCritical EMPTY >
<!ENTITY % ThresholdType "(Absolute | Relative)?" >
<!ELEMENT CalcSingleThreshold (%ThresholdType;,
                               Threshold, SetTo?) >
<!ELEMENT CalcMultiThreshold (%ThresholdType;,
                               Calc*) >
<!ELEMENT Calc ((Warning | Minor | Major | Critical),
               Threshold, SetTo?) >

<!-- Status Propagation Rules -->
<!ELEMENT PropRule (Name, Prop*) >
<!ELEMENT Prop ((%Severity;)?,
               (Increase | Decrease | Unchanged |
                Ignore | SetTo)) >

<!-- Actions -->
<!ELEMENT Action (Name,
                 Label?,
                 Description?,
                 (Program | URL))>

<!ELEMENT Program (Command,
                  User,
                  ServiceNodes?,
                  ServerNode?,
                  LocalNode?,
                  NodeRef*) >
```

```
<!-- Service Elements -->
<!ELEMENT Service (Name,
    Status?,
    Label?,
    Description?,
    Title?,
    Icon?,
    Depth?,
    Background?,
    MsgWeight?,
    CalcRuleRef?,
    MsgPropRuleRef?,
    OriginalId?,
    (NodeRef | Attribute | ActionRef |
    Source | MsgSvcName)*) >

<!-- Service Attributes -->
<!ELEMENT Attribute (Name, Value) >

<!-- Associations -->
<!ELEMENT Source ((Dependency | Composition),
    ServiceRef,
    Status?,
    Label?,
    Weight?,
    PropRuleRef?) >

<!ELEMENT Association ((Dependency | Composition |
    OperatorAssignment),
    SourceRef,
    TargetRef,
    Status?,
    Label?,
    Weight?,
    PropRuleRef?) >

<!-- Operators -->
<!ELEMENT Operator (Name,
    Label?,
    Description?,
    Title?,
    Icon?,
    Depth?,
```

```
Background?,  
ServiceRef*)>
```

```
<!-- Basic Attribute Elements -->  
<!ELEMENT Status          %Severity; >  
<!ELEMENT SetTo           %Severity; >  
  
<!ELEMENT CalcRuleRef     (#ASCII) >  
<!ELEMENT MsgPropRuleRef (#ASCII) >  
<!ELEMENT ServiceRef      (#ASCII) >  
<!ELEMENT SourceRef       (#ASCII) >  
<!ELEMENT TargetRef       (#ASCII) >  
<!ELEMENT PropRuleRef     (#ASCII) >  
<!ELEMENT ActionRef       (#ASCII) >  
<!ELEMENT AttrRef         (#ASCII) >  
<!ELEMENT NodeRef         (#ASCII) >  
<!ELEMENT Name            (#ASCII) >  
<!ELEMENT Label           (#PCDATA) >  
<!ELEMENT Description     (#PCDATA) >  
<!ELEMENT Threshold       (#PCDATA) >  
<!ELEMENT Increase        (#PCDATA) >  
<!ELEMENT Decrease        (#PCDATA) >  
<!ELEMENT MsgWeight       (#PCDATA) >  
<!ELEMENT Weight          (#PCDATA) >  
<!ELEMENT Value           (#PCDATA) >  
<!ELEMENT Command         (#PCDATA) >  
<!ELEMENT URL             (#PCDATA) >  
<!ELEMENT Title           (#PCDATA) >  
<!ELEMENT Background      (#PCDATA) >  
<!ELEMENT Depth           (#PCDATA) >  
<!ELEMENT Icon            (#PCDATA) >  
<!ELEMENT User            (#PCDATA) >  
<!ELEMENT Class           (#PCDATA) >  
<!ELEMENT Width           (#PCDATA) >  
<!ELEMENT Height          (#PCDATA) >
```

The Service Configuration File Syntax

```
<!ELEMENT MsgSvcName      (#PCDATA) >
<!ELEMENT OriginalId      (#PCDATA) >
<!ELEMENT Absolute        EMPTY >
<!ELEMENT Relative        EMPTY >
<!ELEMENT Unchanged       EMPTY >
<!ELEMENT Ignore          EMPTY >
<!ELEMENT Normal          EMPTY >
<!ELEMENT Warning         EMPTY >
<!ELEMENT Minor           EMPTY >
<!ELEMENT Major           EMPTY >
<!ELEMENT Critical        EMPTY >
<!ELEMENT ServerNode      EMPTY >
<!ELEMENT ServiceNodes    EMPTY >
<!ELEMENT LocalNode       EMPTY >
<!ELEMENT Dependency      EMPTY >
<!ELEMENT Composition     EMPTY >
<!ELEMENT OperatorAssignment EMPTY >

<!-- EOF -->
```

The following sections describe the tags used in the service configuration file.

Defining the Services and Associations

The following are descriptions of the service tags. The definition includes the relationship of the service to other services, and the attributes of the service.

- The Root Service Tags
- The Service Tags
- The Source Tags
- The Association Tags
- The Calculation Rule Tags
- The Propagation Rule Tags
- The Service Action Tags
- The Operator Tags

Table B-3 The Root Service Tags

| Tag | Required? | Description |
|---------------|----------------------|--|
| <Services> | Required. | Is the root element. It contains the following tags: <ul style="list-style-type: none">• <CalcRule>• <PropRule>• <Action>• <Service>• <Operator>• <Association> |
| <CalcRule> | Any number possible. | Specifies the calculation rule. |
| <PropRule> | Any number possible. | Specifies the propagation rule. |
| <Action> | Any number possible. | Specifies the action. |
| <Service> | Any number possible. | Specifies the service. |
| <Operator> | Any number possible. | Specifies the operator. |
| <Association> | Any number possible. | Specifies an association. |

Table B-4 The Service Tags

| Tag | Required? | Description |
|---------------|----------------------|---|
| <Service> | Any number possible. | <p>Specifies the service. It contains the following tags:</p> <ul style="list-style-type: none"> • <Name> • <Status> • <Label> • <Description> • <Title> • <Icon> • <Depth> • <Background> • <MsgWeight> • <CalcRuleRef> • <MsgPropRuleRef> • <OriginalId> • <NodeRef> • <Attribute> • <ActionRef> • <Source> • <MsgSvcName> |
| <Name> | Required. | <p>Specifies the name of the service. The name must match the string that is entered in the template, condition, or regroup condition window, or supplied as parameter of <code>opcmsg (1)</code>.</p> <p>Note that the name must not be longer than 254 characters, and must not start with a hyphen (-).</p> |
| <Status> | Optional. | <p>Contains the current status of the service. This tag is ignored when specified in a service configuration file. Status queries, for example with the command <code>opcservice -xml -list</code> or with the Service Navigator APIs of the Developer's Toolkit, output a severity for the service.</p> |
| <Label> | Optional. | <p>Specifies the label of the service. The string that is entered here is displayed in the Service Navigator GUI. If <Label> is not used, <Name> is displayed.</p> |
| <Description> | Optional. | <p>Specifies the description of the service. A detailed description can be useful for future reference.</p> |

Table B-4 The Service Tags (Continued)

| Tag | Required? | Description |
|------------------|----------------------|--|
| <Title> | Optional. | Specifies the title of the service window in the Service Navigator GUI. If <Title> is not used, <Label> is displayed. |
| <Icon> | Optional. | Specifies an icon for the service. |
| <Depth> | Optional. | Specifies the number of hierarchical levels displayed in the Service Navigator GUI. By default, two levels are displayed: the service itself and one more level. The number must be an integer, for example 3. |
| <Background> | Optional. | Specifies a background graphic for the content pane in the Service Navigator GUI. Graphics are defined through URLs or can be called from the filesystem on the management server. If <Background> is not used, a default white background is used. |
| <MsgWeight> | Optional. | Specifies the weight of the service's own messages against the weight of its subservices. The default value is 1. The number can be a floating value, for example 1.8 or 2.5. |
| <CalcRuleRef> | Optional. | References the calculation rule (defined in <CalcRule>) to be applied to this service. |
| <MsgPropRuleRef> | Optional. | References the propagation rule (defined in <PropRule>) for the service's messages. |
| <OriginalId> | Optional. | Identifier set by HP OpenView Service Configuration for Service Navigator. |
| <NodeRef> | Any number possible. | References one or more nodes that belong to a service. |
| <Attribute> | Any number possible. | <p>Specifies name=value pairs which can be referenced by service actions. It contains the following tags:</p> <ul style="list-style-type: none"> • <Name> • <Value> <p>See "Reserved Service Attributes" on page 165 for a list of attribute names that are reserved for Service Navigator internal use.</p> |

Table B-4 The Service Tags (Continued)

| Tag | Required? | Description |
|--------------|----------------------|---|
| <ActionRef> | Any number possible. | Specifies a service action. |
| <Source> | Any number possible. | Specifies a subservice. |
| <MsgSvcName> | Any number possible. | “Service name in message” attribute of the service. Used by HP OpenView Service Configuration for Service Navigator only. |

Table B-5 The Source Tags

| Tag | Required? | Description |
|---------------|----------------------|--|
| <Source> | Any number possible. | Specifies the service. It contains the following tags: <ul style="list-style-type: none"> • <Dependency> • <Composition> • <ServiceRef> • <Status> • <Label> • <Weight> • <PropRuleRef> |
| <Dependency> | Required. | Specifies a usage relationship to a subservice. |
| <Composition> | Required. | Specifies a containment relationship to a subservice. |
| <ServiceRef> | Required. | References the subservice as defined in <Service>. |
| <Status> | Optional. | Contains the current status of the service. This tag is ignored when specified in a service configuration file. Status queries, for example with the command <code>opcservice -list, -xml</code> , or with the Service Navigator APIs of the Developer’s Toolkit, output a severity for the service. |
| <Label> | Optional. | Specifies the name of the link between the service and the subservice. The label is displayed in the Service Navigator GUI. |

Table B-5 The Source Tags (Continued)

| Tag | Required? | Description |
|---------------|-----------|---|
| <Weight> | Optional. | Specifies the factor to be used to modify the importance of a service's effect on its parent. This can be a decimal number. The default value is 1. The number can be a floating value, for example 1.8 or 2.5. |
| <PropRuleRef> | Optional. | References the propagation rule (defined in <PropRule>) for the service's messages. |

Table B-6 The Association Tags

| Tag | Required? | Description |
|-----------------------|----------------------|--|
| <Association> | Any number possible. | <p>Specifies an association of a service to another service. It contains the following tags:</p> <ul style="list-style-type: none"> • <Dependency> • <Composition> • <OperatorAssignment> • <SourceRef> • <TargetRef> • <Status> • <Label> • <Weight> • <PropRuleRef> <p>Using the <Association> tag has an effect similar to using the <Source> tag. The choice depends on your XML coding style. The <Source> tag produces nested XML. The <Association> tag produces linked XML.</p> |
| <Dependency> | Required. | Specifies a usage relationship to a subservice. |
| <Composition> | Required. | Specifies a containment relationship to a subservice. |
| <Operator_Assignment> | Required. | Specifies an assignment of an operator to a service. |
| <SourceRef> | Required. | References the source service as defined in <Service>. |
| <TargetRef> | Required. | References the target service as defined in <Service>. |

Table B-6 The Association Tags (Continued)

| Tag | Required? | Description |
|---------------|------------------|---|
| <Status> | Optional. | Contains the current status of the service. This tag is ignored when specified in a service configuration file. Status queries, for example with the command <code>opcservice -xml -list</code> or with the Service Navigator APIs of the Developer's Toolkit, output a severity for the service. |
| <Label> | Optional. | Specifies the name of the link between the service and the subservice. The label is displayed in the Service Navigator GUI. |
| <Weight> | Optional. | Specifies the factor to be used to modify the importance of a service's effect on its parent. This can be a decimal number. The default value is 1. The number can be a floating value, for example 1.8 or 2.5. |
| <PropRuleRef> | Optional. | References the propagation rule (defined in <PropRule>) for the service's messages. |

Defining Status Calculation and Propagation Rules

The following tags define the status calculation and propagation rules used by Service Navigator.

Table B-7 The Calculation Rule Tags

| Tag | Required? | Description |
|-----------------------|----------------------|---|
| <CalcRule> | Any number possible. | Specifies a calculation rule. It contains the following tags: <ul style="list-style-type: none"> • <Name> • <CalcMostCritical> • <CalcSingleThreshold> • <CalcMultiThreshold> |
| <Name> | Required. | Specifies the name of the calculation rule. |
| <CalcMostCritical> | Required. | Specifies the most critical status calculation rule. This is the default. |
| <CalcSingleThreshold> | Required. | Specifies the single threshold calculation rule. It contains the following tags: <ul style="list-style-type: none"> • <Absolute> • <Relative> • <Threshold> • <SetTo> |
| <CalcMultiThreshold> | Required. | Specifies a multiple threshold calculation rule. It contains the following tags: <ul style="list-style-type: none"> • <Absolute> • <Relative> • <Calc> • <Threshold> • <SetTo> |
| <Absolute> | Optional. | Specifies an absolute value for <Threshold>. |
| <Relative> | Optional. | Specifies a percentage value for <Threshold>. |
| <Threshold> | Required. | Specifies the threshold value. If an absolute threshold is specified, an integer value must be used. If a relative threshold is specified, a floating point value between 0.0 (0%) and 1.0 (100%) must be used. |

Table B-7 The Calculation Rule Tags (Continued)

| Tag | Required? | Description |
|------------|----------------------|---|
| <SetTo> | Optional. | <p>Specifies a severity which the service adopts. Contains one of the following tags:</p> <ul style="list-style-type: none"> • <Normal> • <Warning> • <Minor> • <Major> • <Critical> |
| <Calc> | Any number possible. | <p>Specifies a value for each severity. It contains the following tags:</p> <ul style="list-style-type: none"> • <Warning> • <Minor> • <Major> • <Critical> • <Threshold> • <SetTo> |
| <Normal> | Required. | Specifies the severity normal. |
| <Warning> | Required. | Specifies the severity warning. |
| <Minor> | Required. | Specifies the severity minor. |
| <Major> | Required. | Specifies the severity major. |
| <Critical> | Required. | Specifies the severity critical. |

Table B-8 The Propagation Rule Tags

| Tag | Required? | Description |
|------------|----------------------|---|
| <PropRule> | Any number possible. | <p>Specifies a propagation rule. It contains the following tags:</p> <ul style="list-style-type: none"> • <Name> • <Prop> |
| <Name> | Required. | Specifies the name of the propagation rule. |
| <Prop> | Any number possible. | <p>Specifies the single threshold calculation rule. It contains the following tags:</p> <ul style="list-style-type: none"> • <Normal> • <Warning> • <Minor> • <Major> • <Critical> |
| <Normal> | Optional. | <p>Specifies the severity normal. It contains the following tags:</p> <ul style="list-style-type: none"> • <Increase> • <Decrease> • <Unchanged> • <Ignore> • <SetTo> |
| <Warning> | Optional. | <p>Specifies the severity warning. It contains the following tags:</p> <ul style="list-style-type: none"> • <Increase> • <Decrease> • <Unchanged> • <Ignore> • <SetTo> |
| <Minor> | Optional. | <p>Specifies the severity minor. It contains the following tags:</p> <ul style="list-style-type: none"> • <Increase> • <Decrease> • <Unchanged> • <Ignore> • <SetTo> |

Table B-8 The Propagation Rule Tags (Continued)

| Tag | Required? | Description |
|-------------|-----------|--|
| <Major> | Optional. | Specifies the severity major. It contains the following tags: <ul style="list-style-type: none"> • <Increase> • <Decrease> • <Unchanged> • <Ignore> • <SetTo> |
| <Critical> | Optional. | Specifies the severity critical. It contains the following tags: <ul style="list-style-type: none"> • <Increase> • <Decrease> • <Unchanged> • <Ignore> • <SetTo> |
| <Increase> | Required. | Defines the number of severity levels by which a given severity is increased. |
| <Decrease> | Required. | Defines the number of severity levels by which a given severity is decreased |
| <Unchanged> | Required. | Defines that severity of the subservice is not changed. This is the default. |
| <Ignore> | Required. | Defines that the severity is ignored in terms of status propagation. |
| <SetTo> | Required. | Specifies the severity which the service adopts. Contains one of the following tags: <ul style="list-style-type: none"> • <Normal> • <Warning> • <Minor> • <Major> • <Critical> |

See “What is the Status of a Service?” on page 29 for more information about the concept behind calculation and propagation rules.

Defining Service Actions

The following lines define actions that can be assigned to services. Service actions are started by a popup menu.

Table B-9 The Service Action Tags

| Tag | Required? | Description |
|---------------|----------------------|---|
| <Action> | Any number possible. | Specifies a service action. |
| <Name> | Required. | Specifies the name of the service action. |
| <Label> | Optional. | Specifies the label of the service action. The label is displayed in the popup menu of the Service Navigator GUI. |
| <Description> | Optional. | Specifies the description of the service action. A detailed description can be useful for future reference. |
| <Program> | Required. | <p>Specifies the command to be executed when the service action is started. Note that you can only run the following type of actions:</p> <ul style="list-style-type: none"> • output-only applications • X-applications if they are redirected to the GUI client <p>Contains the following tags:</p> <ul style="list-style-type: none"> • <Command> • <User> • <ServiceNodes> • <ServerNode> • <LocalNode> • <NodeRef |
| <URL> | Required. | Specifies the URL of the web application to be started by the service action. The application is displayed in a separate window. |
| <Command> | Required. | Specifies the command to be executed when the service action is started. |
| <User> | Required. | Specifies the user who executes the service action. Must be specified for <Program> actions. |

Table B-9 The Service Action Tags (Continued)

| Tag | Required? | Description |
|----------------|----------------------|---|
| <ServiceNodes> | Optional. | Executes the service action on the nodes defined in <Service>. |
| <ServerNode> | Optional. | Executes the service action on the OVO management server. |
| <LocalNode> | Optional. | Executes the service action on the computer where the Service Navigator is running. |
| <NodeRef> | Any number possible. | Specifies the nodes on which the service action is to be executed. |

Assigning Services to Operators

The following are descriptions of the operator tags.

Table B-10 **The Operator Tags**

| Tag | Required? | Description |
|---------------|----------------------|--|
| <Operator> | Any number possible. | Specifies an operator-to-service assignment. Contains the following tags: <ul style="list-style-type: none"> • <Name> • <Label> • <Description> • <Title> • <Icon> • <Depth> • <Background • <ServiceRef> |
| <Name> | Required. | Specifies the name of the operator, for example <code>opc_op</code> . |
| <Label> | Optional. | Specifies the label of the toplevel service that is assigned to the operator. The label is displayed in the Service Navigator GUI. |
| <Description> | Optional. | Specifies the description of the service assignment. A detailed description can be useful for future reference. |
| <Title> | Optional. | Specifies the title of the service window in the Service Navigator GUI. If <Title> is not used, <Label> is displayed. |
| <Icon> | Optional. | Specifies an icon for the toplevel service that is being assigned. |
| <Depth> | Optional. | Specifies the number of hierarchical levels displayed in the Service Navigator GUI. By default, two levels are displayed: the service itself and one more level. The number must be an integer, for example 3. |
| <Background> | Optional. | Specifies a background graphic for the content pane in the Service Navigator GUI. Graphics are defined through URLs or can be called from the filesystem on the management server. If <Background> is not used, a default white background is used. |
| <ServiceRef> | Any number possible. | Specifies the service (defined in <Service>) for which the operator is responsible. If <ServiceRef> is not defined, the toplevel service is assigned. |

Reserved Service Attributes

The following attribute names are reserved for internal use by Service Navigator. They are used to achieve special effects in the GUI, for example for enhancing service submaps and for labeling service icons dynamically.

Attribute name: **ov_label<n>**

Description: Defines additional labels for service icons.

<n> determines the position of the label below the service icon. Must be a positive integer.

Attribute value: The value of the attribute can be one of the following:

- <string>
Specifies label text.
- ov_image:<image>
Specifies an image to be displayed as label.
<image> must either be the full path to the image in the file system on the management server or a URL (http or ftp).

Example:

```
<Attribute>
  <Name>ov_label1</Name>
  <Value>First Label</Value>
  <Name>ov_label2</Name>
  <Value>ov_image:/opt/OV/www/htdocs/\
    ito_op/images/second_label.gif</Value>
</Attribute>
```

Attribute name: **ov_background**

Description: Defines an image to be displayed on the background of the submap of a service.

Value: ov_image:<image>
<image> must either be the full path to the image in the file system on the management server or a URL (http or ftp).

Example:
<Attribute>
 <Name>ov_background</Name>
 <Value>ov_image:/etc/opt/OV/share/\n backgrounds/italy.gif</Value>
</Attribute>

Attribute name: **ov_map_width**

Description: Defines the width of a submap background in pixels.

Value: <string>
<string> must be a positive integer. The value is defined in pixels.

Example:
<Attribute>
 <Name>ov_map_width</Name>
 <Value>100</Value>
</Attribute>

Attribute name: **ov_map_height**

Description: Defines the height of a submap background in pixels.

Value: <string>
<string> must be a positive integer. The value is defined in pixels.

Example:
<Attribute>
 <Name>ov_map_height</Name>
 <Value>100</Value>
</Attribute>

Attribute name: **ov_posX**

Description: Defines the position of the service icon on the x axis of the submap of the parent service (in pixels).

Value: `<string>`
`<string>` must be a positive integer. The value is defined in pixels.

Example:
`<Attribute>`
 `<Name>ov_posX</Name>`
 `<Value>50</Value>`
`</Attribute>`

Attribute name: **ov_posY**

Description: Defines the position of the service icon on the y axis of the submap of the parent service (in pixels).

Value: `<string>`
`<string>` must be a positive integer. The value is defined in pixels.

Example:
`<Attribute>`
 `<Name>ov_posX</Name>`
 `<Value>50</Value>`
`</Attribute>`

The opcservice Command

The command `opcservice` adds, replaces, and deletes service configurations. It lets you assign services to operators as part of their responsibilities, and de-assign services if they are no longer responsible. It is located in the following directory:

```
/opt/OV/bin/OpC/opcservice
```

The `opcservice` command interprets the character set of its command line arguments by the locale setting. Therefore, before calling `opcservice`, make sure that the locale set for your terminal window corresponds to the system startup language. You can determine the system startup language by reading the following files:

- ❑ on HP-UX: `/etc/rc.config.d/LANG`
- ❑ on Sun Solaris: `/etc/default/init`

If `LANG` is not set in these files, `C` is used by default.

The output of the `opcservice` command is in the external character set, that is, the character set in which the command was started. The `-xml` option is one exception: if you call `opcservice -xml`, the output is in the external character set UTF-8.

Note that there is also a wrapper `/opt/OV/bin/OpC/opcservice.sh` which is linked to `/usr/bin/opcservice` and first calls `opcsvconv` to convert the configuration files to the new syntax, if necessary.

opcservice has the following command line options (which can also be abbreviated):

```
opcservice
    [-help] |
    [-verbose] |
    [-xml] |
    -list
    [[-services] [-all] [-recursive|-subentity] [-full]]
    [<service_name>...]
    [-actions [<action_name>...]]
    [-calcrules [<calculation_rule_name>...]]
    [-proprules [<propagation_rule_name>]]
    [-operators [<operator>...]] |
    -add <configuration_file>
    [[-services] [<service_name>...]]
    [-actions [<action_name>...]]
    [-calcrules [<calculation_rule_name>...]]
    [-proprules [<propagation_rule_name>]]
    [-operators [<operator>...]] |
    -replace <configuration_file>
    [[-services] [<service_name>...]]
    [-actions [<action_name>...]]
    [-calcrules [<calculation_rule_name>...]]
    [-proprules [<propagation_rule_name>]]
    [-operators [<operator>...]] |
    -remove [[-services] [<service_name>...]]
    [-actions [<action_name>...]]
    [-calcrules [<calculation_rule_name>...]]
    [-proprules [<propagation_rule_name>]]
    [-operators [<operator>...]] |
    -check <configuration_file> |
    -assign <operator> <service_name> [<service_name>...] |
    -deassign <operator> [<service_name>...] |
    -assignments <service_name> [<service_name>...] |
    -operators [<operator>...] |
```

Reference Information
The opcservice Command

```
-services [<service_name>...] |  
-actions [<action_name>...] |  
-proprules [<propagation_rule_name>...] |  
-calcrules [<calculation_rule_name>...] |  
-logs <service_name>[<service_name>...] |  
-log_enable [-recursive|-depth <number>]  
<service_name>[<service_name>...] |  
-log_disable [<service_name>...]  
-dump <file_name>
```

Note that the following options are deprecated:

```
-remove_action  
-remove_calcrule  
-remove_proprule  
-globals  
-subentity
```

Where:

- help Prints usage message for opcservice; all other options are ignored.
- verbose Creates verbose output and displays all messages on standard output.
- xml Causes opcservice to output pure XML format in the code set UTF-8.
- list Lists the services defined in the service repository. If `-recursive` is specified, the services are listed recursively. If `-full` is specified, the referenced objects are listed. `-all` lists the entire configuration.
- add Adds the service input file `<configuration_file>` to the service repository file. A list of objects can be specified.

If an object with the same name already exist, errors are issued and the existing objects remain unchanged.
- replace Replaces existing objects defined in `<configuration_file>` or else adds new objects to the service repository file. Objects can be specified by name.
- remove Removes existing objects defined in `<configuration_file>` the service repository file. A objects can be specified. If nothing is specified, everything is removed.
- check Checks that the XML syntax in `<configuration_file>` is correct and consistent. It also issues warnings for duplicate names.
- assign Assigns the service with `<service_name>` to the operator specified in `<operator>`. This service and all contained/used subservices are then visible to the operator.
- deassign De-assigns the service with `<service_name>` from the operator specified in `<operator>`. The operator can no longer see this service and its subservices. If you do not specify a service, all assigned services are deassigned.

The `opcservice` Command

- `-assignments` Lists all operators who are assigned to the service specified in `<service_name>`. If you do not specify a service name, all operators who have a service assigned are listed.
- `-operators` Lists all assignments to all operators. If you specify an operator, only the assignments of that operator are listed.
- `-services` Lists all services. If you specify a service, only that service is listed.
- `-actions` Lists all actions that are defined in the service configuration file. If you do not specify an action name, all actions are listed.
- `-proprules` Lists all propagation rules that are defined in the service configuration file. If you do not specify a propagation rule, all propagation rules are listed.
- `-calcrules` Lists all calculation rules that are defined in the service configuration file. If you do not specify a calculation rule, all calculation rules are listed.
- `-logs` Lists all services whose status is currently logged in the database.
- `-log_enable` Enables logging of service status changes in the database.
- `-log_disable` Disables logging of service status changes in the database.
- `-dump` Outputs the current content of the service engine in XML format. This is useful for troubleshooting purposes.

Multiple options can be specified for one command; for example, `opcservice -remove - operators -list - operators`, first removes all operators and then attempts to list them. See also the man page `opcservice(1M)`.

OVO Variables

This section lists and defines the variables that can be used with Service Navigator.

OVO Variables in Service Names

See the *OVO Administrator's Reference* or the *OVO Administrator's Guide to Online Information* for a list of variables that can be used in the Service Name fields of the template and condition windows and the Regroup Condition No. window.

OVO Variables in Service Actions

The following variables can be used in the command string when defining a service action in the service configuration file:

`$OPC_SERVICE_LABEL` Returns the label of a service.

`$OPC_SERVICE_MAPPINGS_SVC_COUNT`
Returns the number of “service name in message” properties of a service.

Sample output:

17

`$OPC_SERVICE_MAPPINGS_SVC [n]`
Returns the *n*th “service name in message” of a service.

Sample output:

SAP:applsv02

`$OPC_SERVICE_MAPPINGS_SVCS`
Returns all “service name in message” properties of a service. The names are separated by spaces.

Sample output:

SAP:applsv01 SAP:applsv02

`$OPC_SERVICE_NAME`

Returns the name of the current service.

`$OPC_SERVICE_NODE`

Returns the name of the node service attribute, if set.

`$OPC_SERVICE_ORIGINAL_ID`

Returns the original ID of a service.

Sample output:

node_fred

`$OPC_SERVICE_VALUE[name]`

Returns the value of the service parameter with name <name> for the selected service; returns an empty string if not set.

OVO Variables in URL Definitions

The following variables can be used when defining a URL:

| | |
|------------------|--|
| \$LANG | Returns the language setting of the user who is running a Service Navigator GUI. |
| \$OPC_GUI_CLIENT | Returns the hostname of the client the GUI is currently running on. |
| \$OPC_MGMTSV | Returns the hostname of the OVO management server. |

Service Navigator Man Pages

This section provides a list of all man pages available for Service Navigator. To refer to the man pages, call them from the command line by using: **man <manpagename>**. If you want to print the man pages, use the command:

```
man <manpagename> | col -lb | lp -d printer_name
```

The man pages are also available in HTML format; open your Internet browser at the following URL:

```
http://<management_server>:3443/ITO_MAN
```

where <management_server> is the fully qualified hostname of your management server.

Man Page

Summary

opcservice(1M)

Configure HP OpenView Service Navigator.

opcsvcattr(1M)

Add, change or remove service attributes.

opcsvcconv(1M)

Convert service configuration files of HP OpenView Service Navigator from the proprietary syntax used with version A.05.xx to XML.

opcsvcdwn(1M)

Download service status logs of HP OpenView Service Navigator to a file.

opcsvcterm(1M)

Emulates an interface to HP OpenView Service Navigator which inputs XML into stdin and outputs XML to stdout.

opcsvcupl(1M)

Upload service status logs of HP OpenView Service Navigator into the OVO database.

Service Navigator Product Bundles

The Service Navigator principle bundle is a hierarchical structure made up of associated bundles, products, and filesets. You can view these software components using the `swinstall` GUI of SD-UX (HP-UX only).

Table B-11 ServiceNavigator Software Bundle

| Product | Product | Product |
|-----------|-----------------------------|---|
| OVOPC-SVC | OVOPC-SVC-DOC | OVO Service Navigator English Documentation. |
| | OVOPC-SVC-JDOC ^a | OVO Service Navigator Japanese Documentation. |
| | OVOPC-SVC-EDOC ^a | OVO Service Navigator Spanish Documentation. |
| | OVOPC-SVC-KDOC ^a | OVO Service Navigator Korean Documentation. |
| | OVOPC-SVC-SDOC ^a | OVO Service Navigator Simplified Chinese Documentation. |
| | OVOPC-SVC-ENG | OVO Service Navigator Localized Files-English. |
| | OVOPC-SVC-KOR ^a | OVO Service Navigator Localized Files-Korean. |
| | OVOPC-SVC-SCH ^a | OVO Service Navigator Localized Files-Simplified Chinese. |
| | OVOPC-SVC-JPN ^a | OVO Service Navigator Localized Files-Japanese. |
| | OVOPC-SVC-MGR | OVO Service Navigator Manager. |
| | OVOPC-SVC-SPA ^a | OVO Service Navigator Localized Files-Spanish. |

- a. Can be removed *after* OVO installation if you want to save disk space or if you do *not* need this component.

Glossary

service A customer-based or user-oriented capability provided by one or more hardware or software components within a computing environment (for example, Email, Network Bandwidth, application access).

service attributes Attributes specified for the service by the OVO administrator. Service attributes can be defined in the service configuration file and dynamically added, changed or removed at runtime through the `opcsvcatr` command or through the `opcmsg` message source template. Reserved attributes are used for service labeling and enhanced service submap presentation.

service hierarchy The ordered dependency relationships between services.

service level agreement A business agreement between a service provider and a service consumer that defines the service level objectives (performance, availability, and so on) and the cost for the service.

severity level A property of an object indicating the status of the object. Severity level is based on the impact of events or messages associated with the object.

Smart Plug-In Prepackaged software that installs into a management console and provides management capabilities specific to a given type of business application, database, operating system, or service, such as SAP/R3.

status calculation Process used to determine the status of a service or object based on the status of the source objects that contribute to the status of the service or object. In OVO for Windows, status

calculation determines the status of a service based on the severity level(s) of subservice(s).

status propagation The way that the status of a status source is represented for status calculation of a superordinate object or service.

man pages, 176

Symbols

\$LANG, 175

\$OPC_GUI_CLIENT, 175

\$OPC_MGMTSV, 175

\$OPC_SERVICE_LABEL, 173

\$OPC_SERVICE_NAME, 174

A

action configuration example, 107

actions, 49

activating the service configuration, 66

additional documentation, 16

Adobe Portable Document Format. *See* PDF documentation

assigning

services to operators, 164

assigning services, 71

attributes, 49

additional labels for services, 125

reserved, 165

B

background image

service submap, 128

C

calculation rule configuration example, 104

concepts, 21

Condition No. window, 69

configuration

Service Navigator, 61

configuring

associations, 156

OVO, 67

services, 152, 153, 155

status calculation, 158

status propagation, 158

containment relationship, 26

conventions, document, 11

custom service maps, 48

customizing

backgrounds, 122

icons, 122

D

decrease

status propagation, 31

defining

service actions, 162

de-installation

Service Navigator, 96

Developer's Toolkit documentation, 16

disabling

service logging, 74

displaying

custom service maps, 46

impacted services, 24

root cause, 24

service properties, 49

service submaps, 46

document conventions, 11

documentation, related

additional, 16

Developer's Toolkit, 16

ECS Designer, 16

online, 17

PDFs, 13

print, 14–15

downloading

service logs, 55, 74

DTD

service configuration file, 144

E

ECS Designer documentation, 16

editing the service configuration file, 64

enabling

service logging, 74

enhanced service submaps, 47

Event Correlation Service Designer. *See* ECS Designer documentation

example

action configuration, 107

calculation rule, 104

configuration files, 119

operator assignment, 111

propagation rule, 104

service configuration, 101

service hierarchy, 27

setting labels dynamically, 126

status calculation, 35

status propagation, 32

F

flexible management environments

opcustprocl, 90

G

get impacted services, 23
get root cause, 24
graphs
 service, 45

GUI

 custom service maps, 46
 get impacted services, 24
 get root cause, 24
 main window, 44
 service properties, 49
 service submaps, 46
 starting, 93
 using remotely, 130

H

HP OpenView Event Correlation Service
 Designer. *See* ECS Designer
 documentation

I

ignore
 status propagation, 31
impacted services analysis, 23
increase
 status propagation, 31
installation
 prerequisites, 59
 Service Navigator, 59
 software bundles, 177

K

kernel parameters, 60

L

labels
 example, 126
 images, 125
 services, 51, 124
 setting dynamically for services, 125
 text, 125
logging
 disabling for services, 74
 enabling for services, 74
 services, 54
 up and download, 55, 74

M

man pages
 for Service Navigator, 176
mapping
 messages to services, 42, 67
max_thread_proc, 60
message logging
 opccustproc1, 85
message mapping, 42
 on managed nodes, 42
 on management server, 42
most critical
 status calculation, 34
multiple thresholds
 status calculation, 34

O

online documentation
 description, 17
\$OPC_SERVICE_LABEL variable, 173
\$OPC_SERVICE_MAPPINGS_SVC_COUNT
 T variable, 173
\$OPC_SERVICE_MAPPINGS_SVC[n]
 variable, 173
\$OPC_SERVICE_MAPPINGS_SVCS
 variable, 173
\$OPC_SERVICE_NAME variable, 174
\$OPC_SERVICE_NODE variable, 174
\$OPC_SERVICE_ORIGINAL_ID variable,
 174
\$OPC_SERVICE_VALUE(name) variable,
 174
opccustproc1
 flexible management environments, 90
 message logging, 85
opcmsg
 message source template example, 86
opcmsg(1), 42, 67, 70
opcservice, 168
 -actions, 172
 -add, 66, 171
 -assign, 72, 171
 -check, 65, 171
 -deassign, 171, 172
 -dump, 172
 -help, 171
 -list, 66, 171
 -log_disable, 172
 -log_enable, 74, 172
 -logs, 172

- operators, 172
- remove, 171
- replace, 171
- services, 172
- verbose, 171
- xml, 171
- opcsvattr, 75
- opcsvcdwn, 74
- opcsvcupld, 74
- OpenView Event Correlation Service Designer. *See* ECS Designer documentation
- OpenView Operations. *See* OVO
- OpenView Reporter, 54
- operator assignment example, 111
- operators
 - assigning services, 71, 164
 - capabilities, 71
- OVO variables
 - in service actions, 173
 - in service names, 118
 - in URLs, 175

P

- PDF documentation, 13
- planning a service hierarchy, 62
- Portable Document Format. *See* PDF documentation
- prerequisites for installation, 59
- print documentation, 14–15
- propagation rule configuration example, 104
- properties
 - service, 49

R

- Regroup Condition No. window, 70
- regroup conditions, 70
- related documentation
 - additional, 16
 - Developer's Toolkit, 16
 - ECS Designer, 16
 - online, 17
 - PDFs, 13
 - print, 14–15
- relationship
 - containment, 26
 - usage, 26
- reporting
 - OpenView Reporter, 21, 54
- reserved attributes, 165

- root cause analysis, 24

S

- scheduled outages, 73
- service
 - configuration file, 52
 - custom maps, 48
 - dynamic update, 52
 - graphs, 45
 - properties, 49
 - relationship to subservice, 26
 - status severity, 29
 - submaps, 48
- service actions
 - defining, 162
 - using OVO variables, 173
- service attributes
 - setting, 75
 - syntax, 83
- service configuration example, 101
- service configuration file, 143
 - activating, 66
 - checking the syntax, 65
 - editing, 64
 - example, 65
 - tags, 144
- service graphs, 45
- service hierarchy
 - definition, 26
 - example, 27
 - planning, 62
- service hours, 73
- service logging, 54
 - enabling, 74
- service names
 - defining, 43
 - for message mapping, 42
 - setting in OVO, 67
 - using OVO variables, 118
- Service Navigator
 - configuring, 61, 100
 - de-installing, 96
 - GUI, 21, 44
 - installing, 59
 - starting the GUI, 93
 - upgrading, 59, 94
 - using remotely, 130
- service properties, 49
- service reporting, 54, 74
- service submap

- enhanced, 47
- service submaps, 46
 - background image, 128
 - background size, 128
 - enhancing, 128
 - subservice position, 128
- ServiceNavigator
 - See Service Navigator
- services
 - assigning operators, 164
 - attributes
 - additional labels, 125
 - labeling, 51
 - reserved attributes, 165
 - setting attributes, 109
 - setting attributes with opcsvattr, 75
 - setting attributes with OVO messages, 75
 - setting labels, 124
- setting
 - service attributes, 75
- setto
 - status propagation, 31
- single threshold
 - status calculation, 34
- software bundles, 177
- starting the GUI, 93
- status calculation
 - configuring, 158
 - most critical, 34
 - multiple thresholds, 34
 - rules, 29, 34
 - single threshold, 34
- status propagation
 - configuring, 158
 - decrease, 31
 - ignore, 31
 - increase, 31
 - rules, 29, 31
 - setto, 31
 - unchanged, 31
- status severity, 29
- submaps
 - service, 46
- subservice position
 - service submaps, 128
- syntax
 - checking, 65
 - service configuration file, 144

T

- tags
 - association, 156
 - calculation rules, 158
 - operators, 164
 - propagation rules, 160
 - root service, 152
 - service, 153
 - service action, 162
 - service configuration file, 144
 - source, 155
- threshold for status calculation, 35
- typographical conventions. *See* document conventions

U

- unchanged
 - status propagation, 31
- upgrade
 - Service Navigator, 94
- upgrading
 - from A.05.xx to A.06.00, 59
- uploading
 - service logs, 55, 74
- URLs
 - using OVO variables, 175
- usage relationship, 26

W

- weighting factor, 30
- window
 - Condition No., 69
 - Regroup Condition No., 70
 - Service Navigator, 21