

# HP NonStop ASAP OpenView Smart Plug- In Manual

## Abstract

Abstract\_text\_here

## Product Version

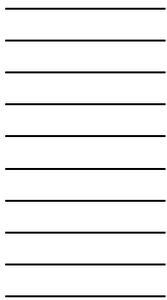
ASAP SE30V2.5

## Supported Release Version Updates (RVUs)

<b>Part Number</b>	<b>Published</b>
000000-000	October 2004

## Document History

Part Number	Product Version	Published
000000-000	Name_and_version_or_NA	Month_Year



# HP NonStop ASAP OpenView Smart Plug-In Manual

<b>Glossary</b>	<b>Index</b>	<b>Examples</b>	<b>Figures</b>	<b>Tables</b>
-----------------	--------------	-----------------	----------------	---------------

- What's New in This Manual iii
  - Manual Information iii
  - New and Changed Information iii
- About This Manual v
  - Notation Conventions v
  - Abbreviations ix

## **1. Introduction to the OpenView Smart Plug-In for ASAP**

- Overview 1-1
- ASAP Data Within OpenView 1-2
  - The OpenView Services Tree 1-2
  - The OpenView Nodes Tree 1-3
  - The OpenView Message Browser 1-5
  - OpenView and OEM State Models 1-7

## **2. Installing the OpenView Smart Plug-In for ASAP**

- System Requirements 2-1
  - Hardware Requirements 2-1
  - Software Requirements 2-1
- Installation Procedure 2-1
  - Installation - Automated 2-2
  - Installation - Manual 2-2
- Verifying the Installation 2-5
- Files Installed 2-7

## **3. Removing the OpenView Smart Plug-In for ASAP**

- Policy Removal Procedure 3-1
- Adapter Removal Procedure 3-2
  - Removal - Automated 3-2
  - Removal - Manual 3-3
- Files Removed 3-4

## **4. Configuring the ASAP OpenView Adapter**

<u>OEM Gateway Adapter Configuration</u>	4-1
<u>Accessing the ASAP OpenView Adapter Configuration Dialog</u>	4-4
<u>OVO Options</u>	4-6
<u>Service and Node Updates</u>	4-7
<u>Manually Updating or Reloading OpenView Data</u>	4-8
<u>Service Population Scheme</u>	4-9
<u>Deleting Services and Nodes From OpenView</u>	4-14
<u>Messaging Options</u>	4-14
<u>Utility Files</u>	4-16
<u>Logging Options</u>	4-16
<u>Enabling and Disabling Logging</u>	4-17
<u>Logging Level</u>	4-18
<u>Log File Name</u>	4-18
<u>Deleting the Log File</u>	4-18
<u>Tracing Options</u>	4-19
<u>Enabling and Disabling Tracing</u>	4-19
<u>Trace Items</u>	4-20
<u>Trace File Name</u>	4-21
<u>Record Limit</u>	4-21
<u>Wrapping Trace Data</u>	4-21
<u>Version Information</u>	4-22
<u>OpenView Adapter Version Data</u>	4-22
<u>Statistics</u>	4-23
<u>Command-Line Options</u>	4-23

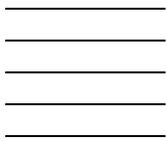
### **Index**

### **Examples**

### **Figures**

<u>Figure 1-1.</u>	<u>OpenView Smart Plug-In for ASAP architecture</u>	1-2
--------------------	---	-----

### **Tables**



# What's New in This Manual

## Manual Information

### Abstract

Abstract\_text\_here

### Product Version

ASAP SE30V2.5

### Supported Release Version Updates (RVUs)

Part Number	Published
000000-000	October 2004

### Document History

Part Number	Product Version	Published
000000-000	Name_and_version_or_NA	Month_Year

## New and Changed Information

This is a new manual.



---

---

---

---

# About This Manual

## Notation Conventions

### Hypertext Links

Blue underline is used to indicate a hypertext link within text. By clicking a passage of text with a blue underline, you are taken to the location described. For example:

This requirement is described under [Backup DAM Volumes and Physical Disk Drives](#) on page 3-2.

### General Syntax Notation

This list summarizes the notation conventions for syntax presentation in this manual.

**UPPERCASE LETTERS.** Uppercase letters indicate keywords and reserved words. Type these items exactly as shown. Items not enclosed in brackets are required. For example:

MAXATTACH

**lowercase italic letters.** Lowercase italic letters indicate variable items that you supply. Items not enclosed in brackets are required. For example:

*file-name*

**computer type.** Computer type letters within text indicate C and Open System Services (OSS) keywords and reserved words. Type these items exactly as shown. Items not enclosed in brackets are required. For example:

myfile.c

**italic computer type.** *Italic computer type* letters within text indicate C and Open System Services (OSS) variable items that you supply. Items not enclosed in brackets are required. For example:

*pathname*

**[ ] Brackets.** Brackets enclose optional syntax items. For example:

TERM [*system-name*.]\$*terminal-name*

INT[ERRUPTS]

A group of items enclosed in brackets is a list from which you can choose one item or none. The items in the list can be arranged either vertically, with aligned brackets on

each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

```
FC [ num ]
   [ -num ]
   [ text ]

K [ X | D ] address
```

**{ }** **Braces.** A group of items enclosed in braces is a list from which you are required to choose one item. The items in the list can be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

```
LISTOPENS PROCESS { $appl-mgr-name }
                  { $process-name }

ALLOWSU { ON | OFF }
```

**|** **Vertical Line.** A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
INSPECT { OFF | ON | SAVEABEND }
```

**...** **Ellipsis.** An ellipsis immediately following a pair of brackets or braces indicates that you can repeat the enclosed sequence of syntax items any number of times. For example:

```
M address [ , new-value ]...

[ - ] {0|1|2|3|4|5|6|7|8|9}...
```

An ellipsis immediately following a single syntax item indicates that you can repeat that syntax item any number of times. For example:

```
"s-char..."
```

**Punctuation.** Parentheses, commas, semicolons, and other symbols not previously described must be typed as shown. For example:

```
error := NEXTFILENAME ( file-name ) ;

LISTOPENS SU $process-name.#su-name
```

Quotation marks around a symbol such as a bracket or brace indicate the symbol is a required character that you must type as shown. For example:

```
"[ repetition-constant-list ]"
```

**Item Spacing.** Spaces shown between items are required unless one of the items is a punctuation symbol such as a parenthesis or a comma. For example:

```
CALL STEPMOM ( process-id ) ;
```

If there is no space between two items, spaces are not permitted. In this example, no spaces are permitted between the period and any other items:

```
$process-name.#su-name
```

**Line Spacing.** If the syntax of a command is too long to fit on a single line, each continuation line is indented three spaces and is separated from the preceding line by a blank line. This spacing distinguishes items in a continuation line from items in a vertical list of selections. For example:

```
ALTER [ / OUT file-spec / ] LINE
      [ , attribute-spec ]...
```

**!i and !o.** In procedure calls, the !i notation follows an input parameter (one that passes data to the called procedure); the !o notation follows an output parameter (one that returns data to the calling program). For example:

```
CALL CHECKRESIZESEGMENT ( segment-id           !i
                        , error                 !o
                        ) ;
```

**!i,o.** In procedure calls, the !i,o notation follows an input/output parameter (one that both passes data to the called procedure and returns data to the calling program). For example:

```
error := COMPRESSEEDIT ( filenum ) ;           !i,o
```

**!i:i.** In procedure calls, the !i:i notation follows an input string parameter that has a corresponding parameter specifying the length of the string in bytes. For example:

```
error := FILENAME_COMPARE_ ( filename1:length  !i:i
                             , filename2:length ) ;  !i:i
```

**!o:i.** In procedure calls, the !o:i notation follows an output buffer parameter that has a corresponding input parameter specifying the maximum length of the output buffer in bytes. For example:

```
error := FILE_GETINFO_ ( filenum           !i
                        , [ filename:maxlen ] ) ;  !o:i
```

## Notation for Messages

This list summarizes the notation conventions for the presentation of displayed messages in this manual.

**Bold Text.** Bold text in an example indicates user input typed at the terminal. For example:

```
ENTER RUN CODE
?123
```

```
CODE RECEIVED:      123.00
```

The user must press the Return key after typing the input.

**Nonitalic text.** Nonitalic letters, numbers, and punctuation indicate text that is displayed or returned exactly as shown. For example:

```
Backup Up.
```

**lowercase italic letters.** Lowercase italic letters indicate variable items whose values are displayed or returned. For example:

```
p-register
process-name
```

**[ ] Brackets.** Brackets enclose items that are sometimes, but not always, displayed. For example:

```
Event number = number [ Subject = first-subject-value ]
```

A group of items enclosed in brackets is a list of all possible items that can be displayed, of which one or none might actually be displayed. The items in the list can be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

```
proc-name trapped [ in SQL | in SQL file system ]
```

**{ } Braces.** A group of items enclosed in braces is a list of all possible items that can be displayed, of which one is actually displayed. The items in the list can be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

```
obj-type obj-name state changed to state, caused by
{ Object | Operator | Service }

process-name State changed from old-objstate to objstate
{ Operator Request. }
{ Unknown. }
```

**| Vertical Line.** A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
Transfer status: { OK | Failed }
```

**% Percent Sign.** A percent sign precedes a number that is not in decimal notation. The % notation precedes an octal number. The %B notation precedes a binary number. The %H notation precedes a hexadecimal number. For example:

```
%005400
%B101111
%H2F
P=%p-register E=%e-register
```

## Notation for Management Programming Interfaces

This list summarizes the notation conventions used in the boxed descriptions of programmatic commands, event messages, and error lists in this manual.

**UPPERCASE LETTERS.** Uppercase letters indicate names from definition files. Type these names exactly as shown. For example:

ZCOM-TKN-SUBJ-SERV

**lowercase letters.** Words in lowercase letters are words that are part of the notation, including Data Definition Language (DDL) keywords. For example:

token-type

**!r.** The !r notation following a token or field name indicates that the token or field is required. For example:

ZCOM-TKN-OBJNAME            token-type ZSPI-TYP-STRING.            !r

**!o.** The !o notation following a token or field name indicates that the token or field is optional. For example:

ZSPI-TKN-MANAGER            token-type ZSPI-TYP-FNAME32.            !o

## Change Bar Notation

Change bars are used to indicate substantive differences between this manual and its preceding version. Change bars are vertical rules placed in the right margin of changed portions of text, figures, tables, examples, and so on. Change bars highlight new or revised information. For example:

The message types specified in the REPORT clause are different in the COBOL85 environment and the Common Run-Time Environment (CRE).

The CRE has many new message types and some new message type codes for old message types. In the CRE, the message type SYSTEM includes all messages except LOGICAL-CLOSE and LOGICAL-OPEN.

## Abbreviations



# 1 Introduction to the OpenView Smart Plug-In for ASAP

The HP OpenView Smart Plug-In for ASAP is a Microsoft Windows-based component that acts as an interface between the HP Open Enterprise Management Gateway (OEM) and HP OpenView Operations for Windows. Its purpose is to integrate ASAP state and performance data on an HP NonStop system into an OpenView management environment.

## Overview

The OpenView Smart Plug-In for ASAP consists of two main components:

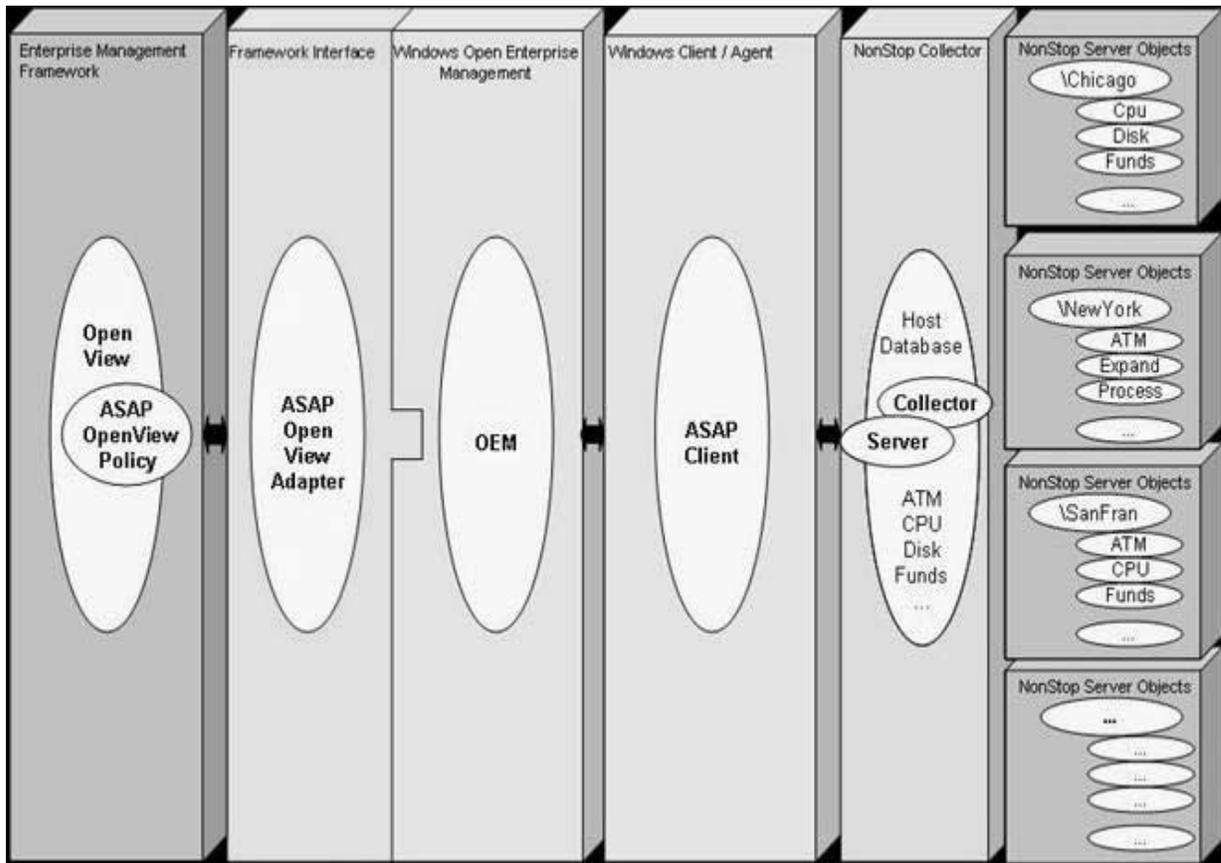
- The ASAP OpenView Adapter, which is a software agent responsible for routing ASAP data from the OEM into OpenView
- An ASAP OpenView policy, which is added to OpenView and interprets and manages ASAP data within OpenView itself

The basic mechanism for communicating data between ASAP and OpenView follows:

- The ASAP components on the NonStop server continually gather state and performance data on both application and system objects.
- This data is sent, on an ongoing basis, to the Windows-based ASAP Client application, which is responsible for displaying the information to the user and performing additional analysis functions.
- Once the ASAP Client has completed its analysis functions, information on all monitored NonStop Kernel objects and their states is sent to the OEM.
- The OEM aggregates and normalizes this data and routes the results to the ASAP OpenView Adapter.
- The Adapter examines the data to determine if it is necessary to update any OpenView data. If so, the relevant changes are sent to OpenView.
- The ASAP OpenView policy within OpenView interprets the changes and updates OpenView data accordingly.

This following diagram depicts the architecture of this environment:

**Figure 1-1. OpenView Smart Plug-In for ASAP architecture**



VST001.vsd

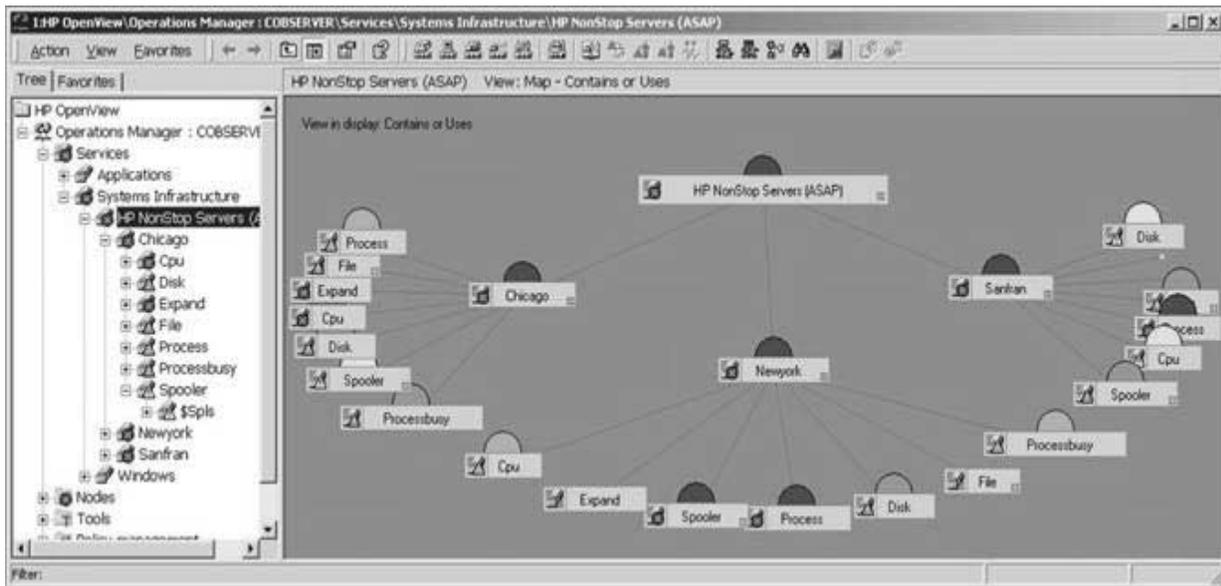
## ASAP Data Within OpenView

Whenever the OpenView Smart Plug-In for ASAP receives new ASAP data, it updates OpenView as necessary. These updates can potentially occur in three different areas of OpenView itself: the Services tree view, the Nodes tree view, and/or the Message Browser.

### The OpenView Services Tree

ASAP data is added to the OpenView Services tree view whenever you begin monitoring a new set of NonStop Kernel objects (for example, you start monitoring a new subsystem such as RDF or the Spooler, add new application domains via ASAPX, or add a new node to your ASAP configuration). All ASAP data in the Services tree

resides under HP NonStop Servers (ASAP) in the System Infrastructure category of services.



VST002.vsd

The level of detail added to the tree view depends on how you configure the Adapter. Choices range from adding nothing to adding every monitored NonStop Kernel object. (For more details on how to configure this option, see the Service Population Scheme description in section 4 .) Regardless of how much detail you add, the data is always organized hierarchically. The level immediately below HP NonStop Servers (ASAP) contains a list of monitored NonStop systems. Each of those systems can contain a list of monitored service types, including system services such as CPU, Disk, and TMF, and also application services monitored via the ASAPX product. Each of those service types can contain a list of subservices, each of which can contain a list of its subservices, and so on, to any level of detail. This organization makes it easy to categorize and view NonStop Kernel objects.

ASAP object state information is automatically propagated within the OpenView Services tree, so you can quickly determine, at a high level, the state of all services in your NonStop network. Once you detect a problem, you can drill down through the hierarchy to find which objects are reporting the error. As you drill down, the OpenView Message Browser reflects only these messages reported against the currently selected object, saving you time in finding the right message.

## The OpenView Nodes Tree

The OpenView Smart Plug-In for ASAP adds a list of all monitored NonStop systems to the OpenView Nodes tree, in the HP NonStop Servers (ASAP) folder. As you add new

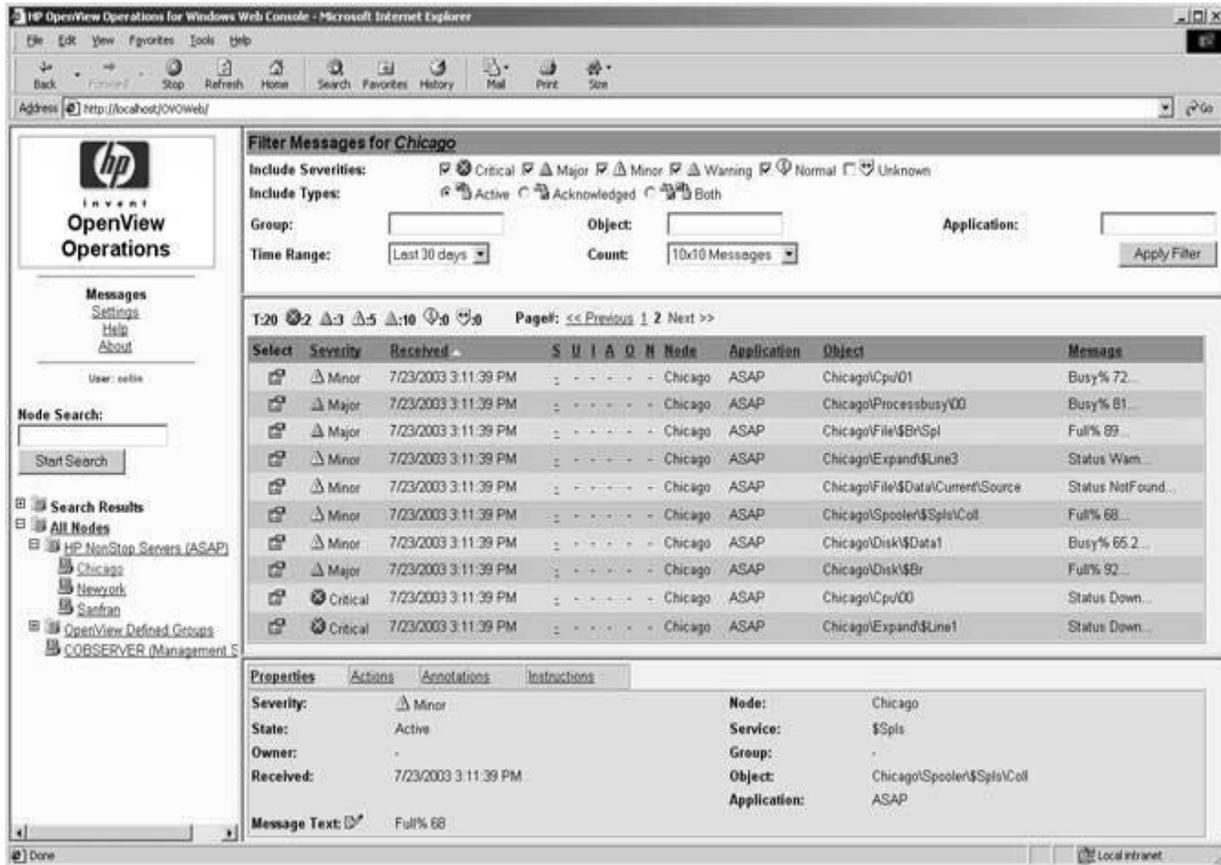
NonStop systems and begin monitoring them, they are automatically added to the tree as well.



VST003.vsd

Unlike the Services view, the Nodes view does not contain a list of services on each system. However, you can still view all messages for objects on a given system by selecting that system in the Nodes tree view. In addition, the Nodes tree view also

appears in the OpenView web browser interface so you can view ASAP data from any web browser on your network.



VST004.vsd

## The OpenView Message Browser

The OpenView Message Browser contains a list of messages logged against objects managed via OpenView. The OpenView SPI for ASAP utilizes this facility to notify you whenever a monitored NonStop Kernel object changes state (for example, a CPU getting busy, a line handler being brought up, or an ATM running low on cash) by logging a message against that object. The details of the message include the name

of the object, the service it is part of, the node it resides on, the severity level, and a text description of the specific error.

Severity	S	U	I	A	O	N	Recei...	Service	Node	Applicatio	Object	Text
Critical	-	-	-	-	-	-	7/23/2003 3:11:3...	Process	Sanfran	ASAP	Sanfran\Process\Zzoos	Status Down
Critical	-	-	-	-	-	-	7/23/2003 3:11:3...	\$Spl	Newyork	ASAP	Newyork\Spooler\Spl\Dev	Status Error
Critical	-	-	-	-	-	-	7/23/2003 3:11:3...	Process	Newyork	ASAP	Newyork\Process\Sgpk	Status Down
Critical	-	-	-	-	-	-	7/23/2003 3:11:3...	Process	Newyork	ASAP	Newyork\Process\Sgpj	Status Down
Critical	-	-	-	-	-	-	7/23/2003 3:11:3...	Expand	Chicago	ASAP	Chicago\Expand\$Line1	Status Down
Critical	-	-	-	-	-	-	7/23/2003 3:11:3...	Cpu	Chicago	ASAP	Chicago\Cpu\00	Status Down
Major	-	-	-	-	-	-	7/23/2003 3:11:3...	\$Spl	Sanfran	ASAP	Sanfran\Spooler\Spl\Coll	Full% 90
Major	-	-	-	-	-	-	7/23/2003 3:11:3...	Work	Sanfran	ASAP	Sanfran\File\$Tools\Work\...	Full% 93
Major	-	-	-	-	-	-	7/23/2003 3:11:3...	Disk	Newyork	ASAP	Newyork\Disk\$Br	Full% 80
Major	-	-	-	-	-	-	7/23/2003 3:11:3...	Processbusy	Chicago	ASAP	Chicago\Processbusy\00	Busy% 81
Major	-	-	-	-	-	-	7/23/2003 3:11:3...	\$Br	Chicago	ASAP	Chicago\File\$Br\Spl	Full% 89
Major	-	-	-	-	-	-	7/23/2003 3:11:3...	Disk	Chicago	ASAP	Chicago\Disk\$Br	Full% 92
Minor	-	-	-	-	-	-	7/23/2003 3:11:3...	Processbusy	Sanfran	ASAP	Sanfran\Processbusy\00	Busy% 62
Minor	-	-	-	-	-	-	7/23/2003 3:11:3...	Disk	Sanfran	ASAP	Sanfran\Disk\$Data1	Busy% 65.2
Minor	-	-	-	-	-	-	7/23/2003 3:11:3...	Cpu	Sanfran	ASAP	Sanfran\Cpu\00	Busy% 61
Minor	-	-	-	-	-	-	7/23/2003 3:11:3...	Disk	Newyork	ASAP	Newyork\Disk\$Data1	Busy% 65.2
Minor	-	-	-	-	-	-	7/23/2003 3:11:3...	\$Spl	Chicago	ASAP	Chicago\Spooler\Spl\Coll	Full% 68
Minor	-	-	-	-	-	-	7/23/2003 3:11:3...	Current	Chicago	ASAP	Chicago\File\$Data\Current...	Status NotFou
Minor	-	-	-	-	-	-	7/23/2003 3:11:3...	Expand	Chicago	ASAP	Chicago\Expand\$Line3	Status Warn

VST005.vsd

As you select a particular NonStop node in the Nodes tree view, or a NonStop service in the Services tree view, the browser shows only messages relevant to the selected object. For example, the contents of the Message Browser for the service CPU on system Chicago:

Severity	S	U	I	A	O	N	Received	Service	Node	Applicatio	Object	Text
Critical	-	-	-	-	-	-	7/23/2003 3:11:39 PM	Cpu	Chicago	ASAP	Chicago\Cpu\00	Status Down
Minor	-	-	-	-	-	-	7/23/2003 3:11:39 PM	Cpu	Chicago	ASAP	Chicago\Cpu\01	Busy% 72
Warning	-	-	-	-	-	-	7/23/2003 3:11:40 PM	Cpu	Chicago	ASAP	Chicago\Cpu\02	Busy% 51
Warning	-	-	-	-	-	-	7/23/2003 3:12:37 PM	Cpu	Chicago	ASAP	Chicago\Cpu\03	Busy% 31

VST006.vsd

In addition, all standard OpenView features are available for messages posted against NonStop objects. You can acknowledge a message, assign an owner, add notes, and so on.

## OpenView and OEM State Models

The ASAP Open Enterprise Management Gateway (OEM) allows any one of nine different states to be assigned to an object that is added to its configuration. However, OpenView utilizes a five state model. In addition to its other duties, the ASAP OpenView Adapter also maps OEM states to OpenView states when it logs a message against a particular object. This mapping is ordered from least to most severe:

<b>OEM Numeric State</b>	<b>OEM State Description</b>	<b>OpenView State</b>
1	Exists, no state information known	Normal
2	Up or OK	Normal
3	Low Usage	Warning
4	Medium Usage	Warning
5	High Usage	Warning
6	Warning	Minor
7	Critical	Major
8	Down	Critical
9	State Degradation	Critical

OEM and OpenView each utilize distinct state models, so OEM states cannot simply be passed through to OpenView. The Adapter addresses this issue by automatically mapping a given OEM numeric state to the OpenView state of closest corresponding severity whenever it posts a message against an object.



# Installing the OpenView Smart Plug-In for ASAP

The OpenView Smart Plug-In for ASAP consists of two components:

- The ASAP OpenView Adapter
- An ASAP OpenView policy

The Adapter runs on the OpenView server as a standalone process, and the policy is installed into OpenView itself. Both components must be installed in order for the Smart Plug-In to function as intended.

## System Requirements

### Hardware Requirements

The OpenView Smart Plug-In for ASAP runs on the OpenView server, so its hardware requirements are identical to those of OpenView. To determine what hardware resources are necessary to run OpenView in your environment, consult the documentation for your version of OpenView.

### Software Requirements

In order to run the OpenView Smart Plug-In for ASAP on your OpenView server, this following software must be installed on the server:

- Microsoft Windows 2000 Server or Windows 2003 Server
- HP OpenView Operations for Windows version 7.x
- ASAP Client, any version
- ASAP Open Enterprise Management Gateway (OEM), any version

In addition, OpenView itself might require additional software components. To determine if any such requirements exist, see the documentation for your version of OpenView.

## Installation Procedure

The procedure for installing the Smart Plug-In for ASAP on your OpenView server depends on whether you use automated installation or manual installation. In either case, the software requirements are the same. So even if you install the SPI using the manual setup, you still must have the ASAP Client, OEM, and other components installed on the server in order to use the Smart Plug-In.

The Smart Plug-In for ASAP installation utilizes portions of OpenView in order to load the Smart Plug-In policy into the OpenView environment. The ASAP OpenView policy cannot be installed if OpenView is not present on the server. For that reason, perform the Smart Plug-In installation after you install OpenView. If you cannot do so, then once you do install OpenView on the server, you will need to either reinstall the Smart Plug-In, or else manually install the policy. (See Manual installation ).

## Installation - Automated

The preferred method for installing the Smart Plug-In for ASAP is to use the automated setup wizard. This wizard prompts you for all necessary information and installs the Smart Plug-In for ASAP components automatically.

To perform an automated installation of the Smart Plug-In for ASAP:

1. Open Windows Explorer.
2. Launch the Smart Plug-In for ASAP automated setup wizard.
  - a. If you are installing from the Smart Plug-In for ASAP CD, navigate to the CD drive (that is, D:\) and double-click **Setup.exe**.
  - b. If you are installing a downloaded version of the SPI, navigate to the folder where you saved the downloaded file and double-click **SPIForASAPSetup.exe**.
  - c. If you have previously installed the ASAP Client version 2.5 or later, navigate to the ASAPOpenViewSetup folder contained in the ASAP Client's main installation path. Typically this folder is Tandem\ASAPOpenViewSetup within your *program-files* folder (that is, C:\Program Files\Tandem\ASAPOpenViewSetup), but it might be different if the ASAP Client's installation location was changed. Once you navigate to the correct folder, double-click **SPIForASAPSetup.exe**.
3. Follow the on-screen instructions to complete the installation.
4. To verify that all components were installed correctly, follow the steps described in Verifying The Installation on page blah.

## Installation - Manual

If none of the automated means described previously are available to you, you can install the Smart Plug-In for ASAP on your server manually. For example, you might need to repair the Smart Plug-In installation because it has become corrupted in some way. Rather than attempting to remove and reinstall the Smart Plug-In, you can simply

repair the Smart Plug-In installation itself. To do so, you need to have access to the Smart Plug-In component files:

OVOAdapter.exe	OVOAdapter.chm	OVOAdapter.ico
HHActiveX.dll	HP OpenView Smart Plug-In For ASAP Reference Guide.doc	HP OpenView Smart Plug-In For ASAP Reference Guide.pdf
ReadMe.txt	ASAP.mm	ASAP-opcmsg.policy

These files were installed as part of any prior setup and are also located on the Smart Plug-In for ASAP CD. In addition, you must have access to the OpenView utility program ovpmutil.exe. Typically, this program is installed into the HP OpenView bin folder (C:\Program Files\HP OpenView\bin), and this folder is automatically added to the Windows search path. However, if ovpmutil.exe is not present, or the path has not been set correctly, you will need to obtain the utility and provide its fully qualified path name in the steps next.

To install the Smart Plug-In manually (or to repair a previous installation):

1. Copy the Adapter component files OVOAdapter.exe, OVOAdapter.chm, OVOAdapter.ico, HP OpenView Smart Plug-In For ASAP Reference Guide.doc, HP OpenView Smart Plug-In For ASAP Reference Guide.pdf, and ReadMe.txt to the destination program folder on the server. Typically it is Tandem\SPI for ASAP folder within your *program-files* folder (C:\Program Files\Tandem\SPI for ASAP), but it can be any folder you choose.
2. Within the destination folder, create a folder named Policy (C:\Program Files\Tandem\SPI for ASAP\Policy).
3. Copy the Smart Plug-In policy files ASAP.mm and ASAP-opcmsg.policy to the new folder.
4. Copy the file HHActiveX.dll to the Windows\System32 folder (C:\Windows\System32).
5. Register the Adapter. To do so, open a Command Prompt window and navigate to the destination folder from Step 1 (C:\Program Files\Tandem\SPI for ASAP). Then enter:

```
OVOAdapter -RegServerStatus
```

Press the Enter key. A dialog box confirms that the Adapter has been registered:



VST007.vsd

6. Load the SPI policy into OpenView. To do so, using the Command Prompt window from Step 5, enter:

```
ovpmutil cfg pol upl "Policy\ASAP.mm"
```

Press the Enter key. (The quotation marks are required in the command.) This message confirms that the policy has successfully been uploaded to OpenView:

```
Pmad connected!
Done!
Operation completed successfully.
```

7. Deploy the SPI policy on your server. To do so, using the Command Prompt window from Step 5, enter:

```
ovpmutil dep /p "\SPI for ASAP\ASAP-opcmsg" /n your-
server-name-within-OpenView
```

Press the Enter key. (The quotation marks are required in the command.) This command requires you to supply the OpenView name of your server. In most cases, it is the name of the server followed by (Management Server); for example, OVOServ1 (Management Server). To obtain the name, start an OpenView Console and locate your server in the Nodes tree view. Supply this name; for example:

```
ovpmutil dep /p "\SPI for ASAP\ASAP-opcmsg" /n "OVOServ1
(Management Server)"
```

When the operation is complete, OpenView provides this message:

```
Pmad connected!
Deploying Ver: "1.0" Type:"Open Message Interface"
```

8. Register the HHActiveX.dll file. To do so, using the Command Prompt window from above, navigate to the Windows\System32 folder (C:\Windows\System32) and enter:

```
regsvr32 HHActiveX.dll
```

Press the Enter key. A dialog box confirms that the DLL has been registered.

9. To verify that all components were installed correctly use the steps described in [Verifying the Installation](#).

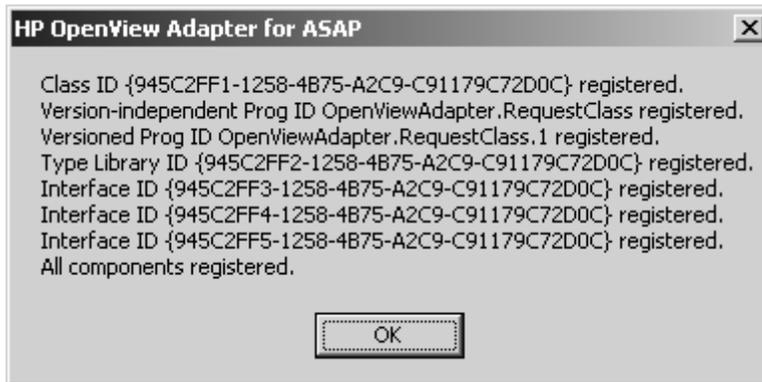
## Verifying the Installation

Once you complete the installation of the OpenView Smart Plug-In for ASAP, confirm that each component was installed correctly:

1. Check that all files are in the correct locations. For the list of files and locations, see Files Installed **blah blah**.
2. Check the Adapter registration. To do so, open a Command Prompt window and navigate to the folder where you installed the Smart Plug-In (C:\Program Files\Tandem\SPI for ASAP). Then enter:

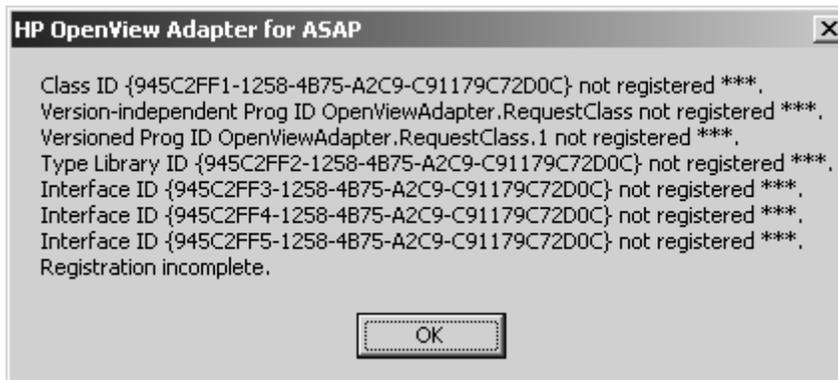
```
OVOAdapter -CheckReg
```

Press the Enter key. If the Adapter is correctly registered on the server, this dialog box appears:



VST008.vsd

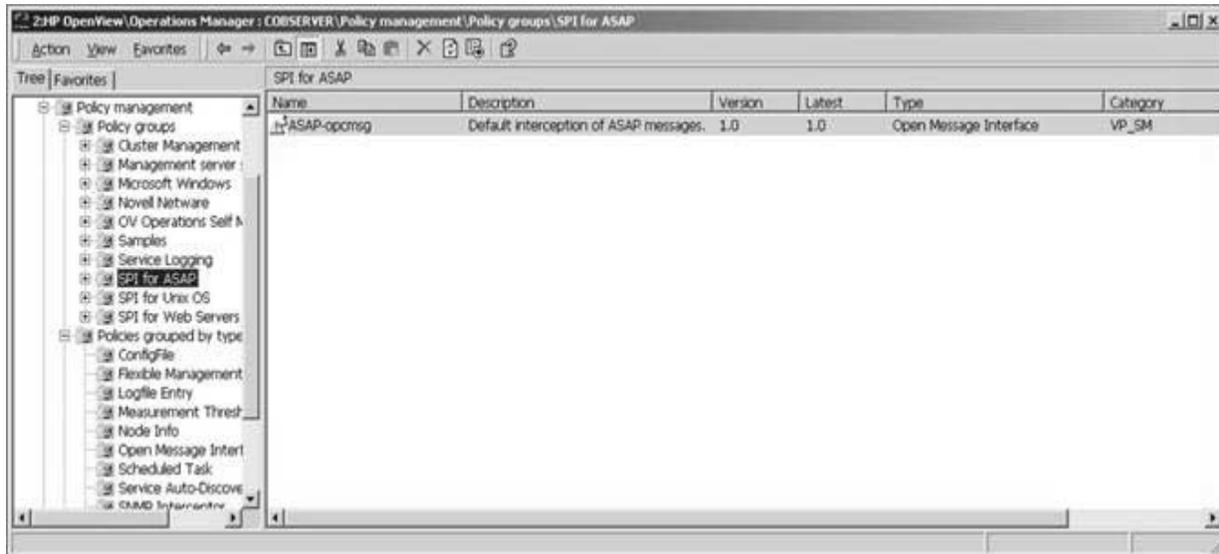
If a problem occurred with the registration, the dialog box indicates which portions are incorrect. This dialog box shows a registration in which all portions are incomplete:



VST009.vsd

To correct registration problems, reregister the Adapter. For details on how to perform the registration, see Step 5 in [Installation - Manual](#) on page 2-2.

3. Verify that the Smart Plug-In policy has been loaded correctly. To do so, start an OpenView Console and expand the Policy management folder in the tree. Then expand the Policy groups folder, and check that the SPI for ASAP policy folder is present:

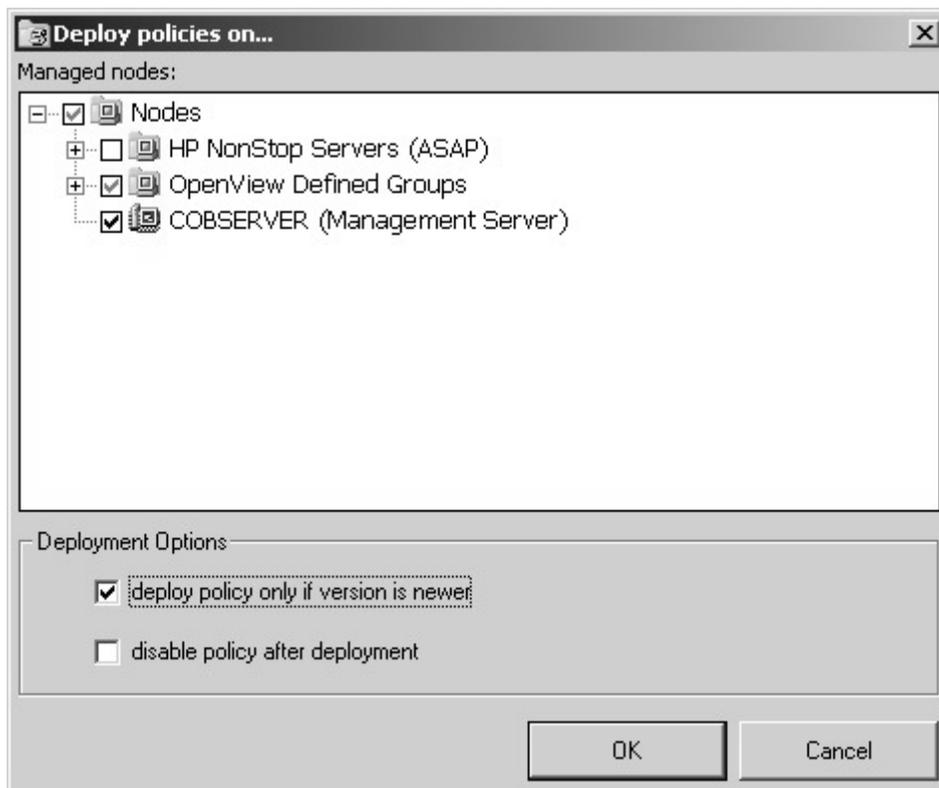


VST010.vsd

4. Verify that the Smart Plug-In policy has been deployed correctly. To do so, select the SPI for ASAP policy folder.

In the right pane of the Console window, right-click the ASAP-opcmg policy and select All Tasks...Deploy on. The Deploy policies on... dialog appears. The box

next to your OpenView server should be selected, indicating that the policy is deployed on the server:



VST011.vsd

If all of the preceding steps are correct, then the OpenView Smart Plug-In for ASAP has been completely installed on your server.

## Files Installed

Regardless of which installation method you choose, this same set of files is installed on the OpenView server:

File Name	Installed To
HP OpenView Smart Plug-In For ASAP Reference Guide.doc	Program directory
HP OpenView Smart Plug-In For ASAP Reference Guide.pdf	Program directory
OVOAdapter.chm	Program directory
OVOAdapter.exe	Program directory
OVOAdapter.ico	Program directory
ReadMe.txt	Program directory

**File Name**

ASAP.mm

ASAP-opcmsg.policy

HHActiveX.dll

**Installed To**

Program directory\Policy

Program directory\Policy

Windows\System32  
directory

---

---

# 3

---

---

## Removing the OpenView Smart Plug-In for ASAP

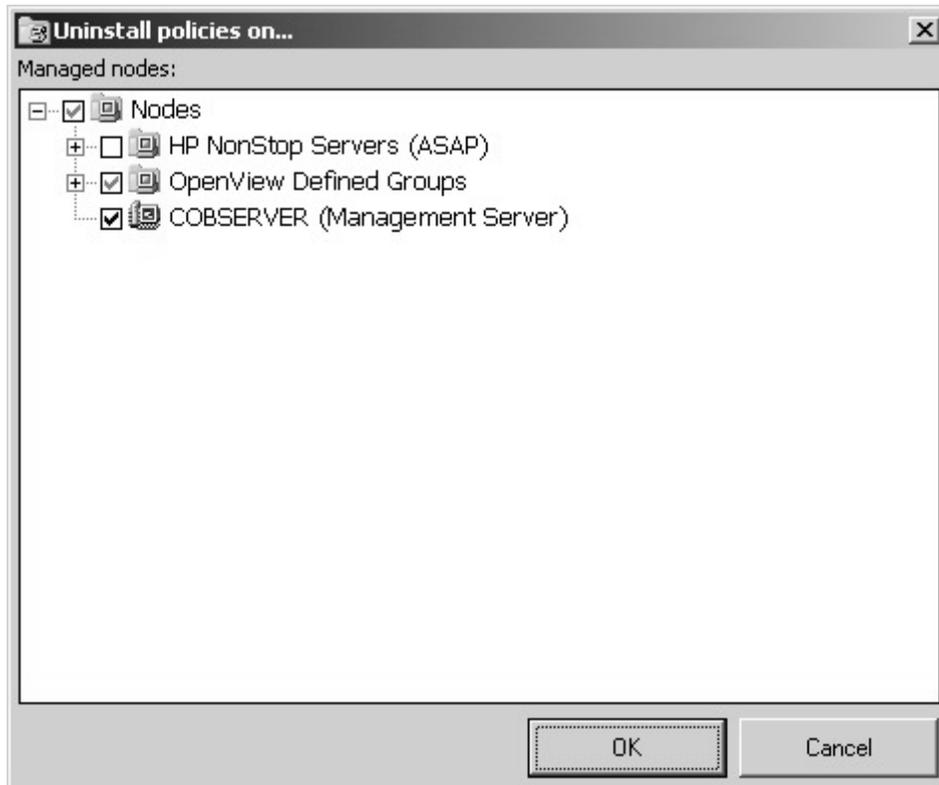
If you want to remove the OpenView Smart Plug-In for ASAP from your server, you must remove both the ASAP OpenView Adapter and the ASAP OpenView policy. To do so, follow the steps outlined in this section. The order does not matter in which you remove the components.

### Policy Removal Procedure

You must remove the ASAP OpenView policy from your OpenView environment via the OpenView Console in two steps: 1. Uninstall the policy from the active OpenView environment. 2. Delete the policy files from the server.

To remove the policy:

1. Start an OpenView Console and expand the Policy management folder.
2. Expand the Policies grouped by type folder, and select Open Message Interface in the tree.
3. Uninstall the policy from the server. In the right pane of the Console window, right-click the ASAP-opcmmsg policy choose All Tasks...Uninstall from.... The Uninstall policies on... dialog box appears. Select the box next to your OpenView server to indicate that the policy should be uninstalled from that system. Then click **OK** to uninstall the policy from your server:



VST012.vsd

4. Delete the policy from the server. In the right pane of the Console window, right-click the ASAP-opcmsg policy, but this time select All Tasks...Delete from server to remove the policy from the server.

## Adapter Removal Procedure

The method you use to remove the ASAP OpenView Adapter from your OpenView server depends on the method you used to originally install the Smart Plug-In for ASAP. In general, always use the removal procedure that corresponds to the initial installation procedure.

### Removal - Automated

If you installed the Adapter using the automated setup wizard, you can remove it using the standard Windows Add/Remove Programs facility:

1. On your server, click the Start button and select Control Panel. The Control Panel window opens.
2. In the Control Panel window, double-click Add/Remove Programs. The Add/Remove Programs dialog box opens.

3. In the Add/Remove Programs dialog box, locate the HP OpenView SPI for ASAP item in the Currently installed programs list, select it, and click Remove.

The ASAP OpenView Adapter will be removed from the server.

## Removal - Manual

If you initially installed the ASAP OpenView Adapter using the manual installation method, you can remove it manually.

To manually remove the Adapter:

1. Delete Adapter configuration information. This step is optional. If you are removing the Adapter, but you plan to reinstall it at some point and want to retain any user-configured Adapter options, skip to Step 2. If you do not plan to reinstall or do not want to retain user-configured option settings, open a Command Prompt window and navigate to the Adapter program folder (C:\Program Files\Tandem\SPI for ASAP). Enter:

```
OVOAdapter -CleanRegStatus
```

Press the Enter key to remove all configuration data from the system. This message appears:



VST013.vsd

2. Delete Adapter registration information. To do so, using the Command Prompt window, enter:

```
OVOAdapter -UnRegServerStatus
```

Press the Enter key to remove all Adapter registration information from the system. This message appears:



VST014.vsd

3. Delete HHActiveX.dll file registration information and the HHActiveX.dll file. (This shared file might be used by other applications, so continue with this step only if you are certain no other applications require it. If you are unsure, skip to Step 4.) To do so, using the Command Prompt window, navigate to the Windows\System32 folder (C:\Windows\System32) and enter:

```
regsvr32 /u HHActiveX.dll
```

Press the Enter key. A dialog box confirms that the registration information for the DLL has been removed. Then delete the HHActiveX.dll file itself.

4. Delete Adapter files. To do so, exit the Command Prompt window. From Windows Explorer, remove the Adapter program folder (C:\Program Files\Tandem\SPI for ASAP) from the system.

## Files Removed

The list of files removed from the server matches those installed:

File Name	Removed From
HP OpenView Smart Plug-In For ASAP Reference Guide.doc	Program directory
HP OpenView Smart Plug-In For ASAP Reference Guide.pdf	Program directory
OVOAdapter.chm	Program directory
OVOAdapter.exe	Program directory
OVOAdapter.ico	Program directory
ReadMe.txt	Program directory
ASAP.mm	Program directory\Policy
ASAP-opcmmsg.policy	Program directory\Policy
HHActiveX.dll	Windows\System32 directory

---

---

# 4

---

---

## Configuring the ASAP OpenView Adapter

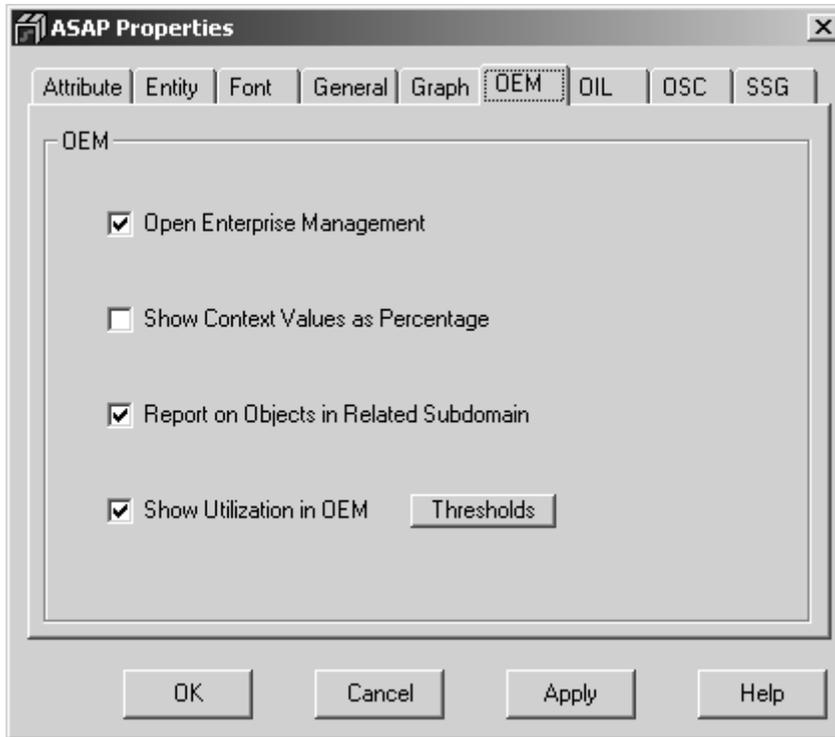
Once you have installed the HP OpenView Smart Plug-In for ASAP on your OpenView server, you need to activate the ASAP OpenView Adapter within the OEM in order to utilize it. In addition, you might also want to tailor how the Adapter interacts with your OpenView environment. The next sections describe how to activate and configure the Adapter.

### OEM Gateway Adapter Configuration

The ASAP subsystem furnishes data to the ASAP OpenView Adapter and therefore OpenView through the ASAP Open Enterprise Management Gateway (OEM). In order for the OEM to send data to the Adapter, you must first activate the Adapter within the OEM's configuration:

1. On your OpenView server, start the ASAP Client.
2. From the Client's main window, click **View...Options**. The ASAP Properties dialog box appears.
3. Click the OEM tab in the dialog box.

4. Select the Open Enterprise Management check box if it is not already selected:



VST015.vsd

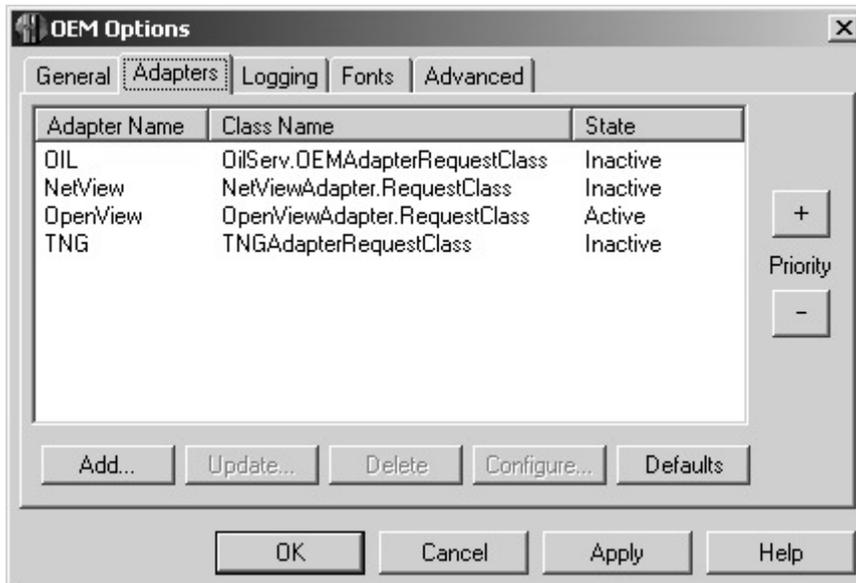
5. Click **OK** to save the change and close the dialog.
6. At this point, the OEM icon should be visible in the system tray of the server (i.e. next to the time):



VST016.vsd

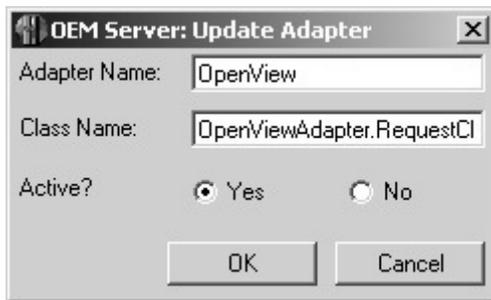
7. Double-click the OEM icon to display the OEM Options dialog box.

- Click the Adapters tab in the dialog box. This tab contains a list of all defined adapters and their current states:



VST017.vsd

- Determine if the OpenView adapter is already correctly defined on your system. This is the case if an entry in the list has an Adapter Name of OpenView, and a Class Name of OpenViewAdapter.RequestClass.
- If the OpenView adapter is defined on your system, double-click that entry in the list to bring up the OEM Server: Update Adapter dialog. To activate the adapter, select **Yes** next to the Active? item, and click **OK**:



VST018.vsd

- If the OpenView adapter is not defined on your system, click **Add...** on the Adapters tab to add a new adapter. The OEM Server: Add Adapter dialog box appears. In the Adapter Name box, enter OpenView. In the Class Name box, enter

OpenViewAdapter.RequestClass. Select **Yes** next to the Active? item, and click **OK** to add the new adapter:



VST019.vsd

12. Click **OK** in the OEM Options dialog box to save these changes and close the dialog box.

The ASAP OpenView Adapter is now active. The OEM will route any ASAP data to the Adapter.

## Accessing the ASAP OpenView Adapter Configuration Dialog

The ASAP OpenView Adapter does not generally provide any kind of visual interface when it is running. You access the Adapter's configuration options through the OEM.

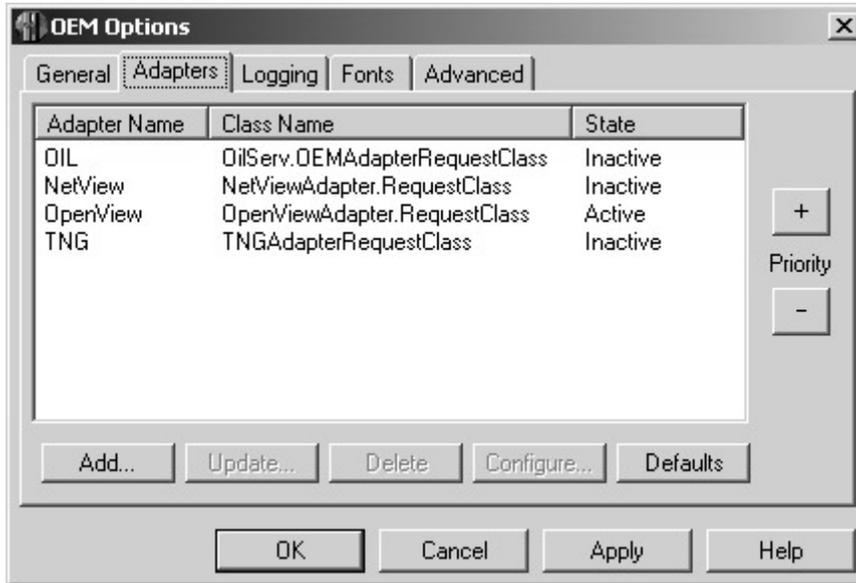
To view and/or change Adapter settings:

1. Double-click the OEM icon in the system tray to display the OEM Options dialog:



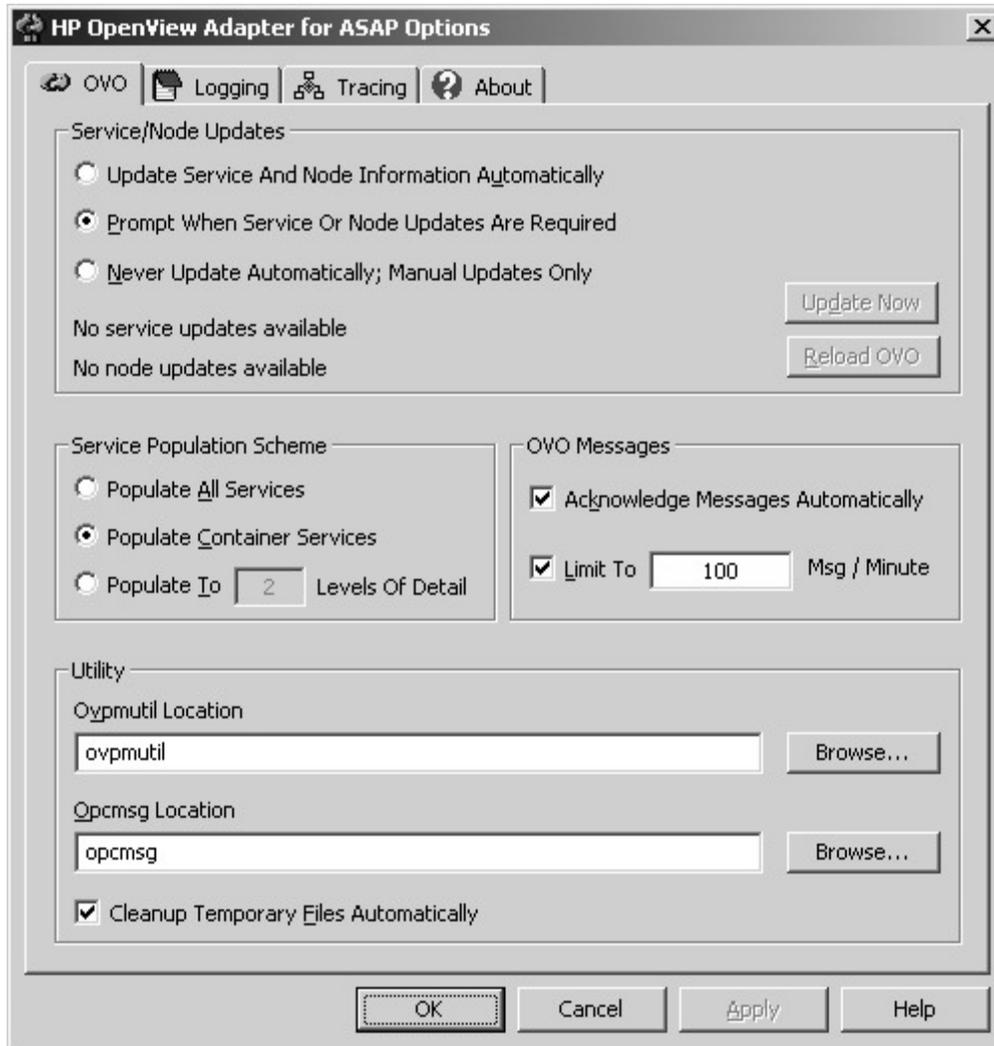
VST020.vsd

2. Click the Adapters tab in the dialog box. This tab contains a list of all defined adapters and their current states:



VST021.vsd

3. Click the **OpenView** entry in the list, and then click **Configure...** .
4. The HP OpenView Adapter For ASAP Options dialog box appears. Change any options you want, as described in the next sections. Click **Apply** to save these changes and leave the dialog box open, click **OK** to save the changes and close the dialog box, or click **Cancel** to close the dialog box without saving changes:



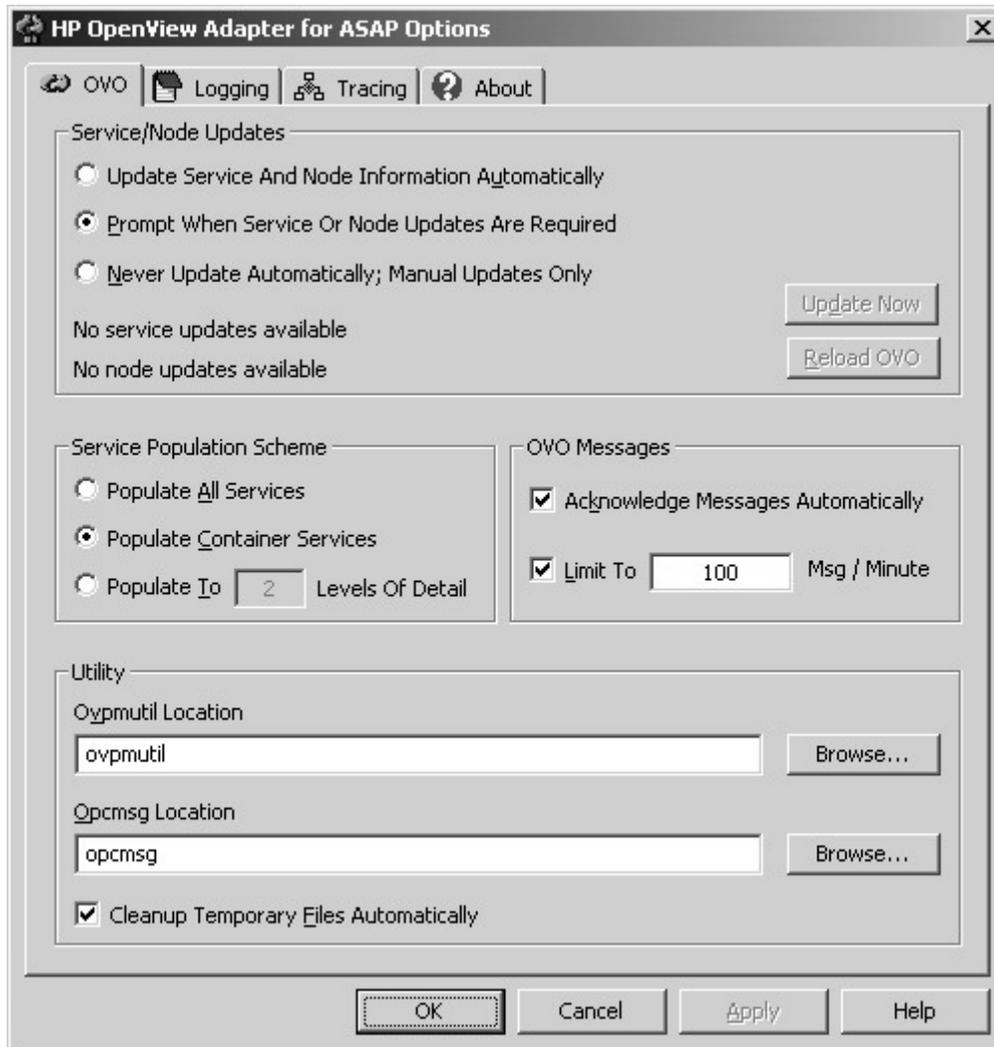
VST022.vsd

5. Click **OK** or **Cancel** in the OEM Options dialog box to close that dialog box. The ASAP OpenView Adapter settings are saved separately from the OEM options, so you need not click **OK** or **Apply** in the OEM Options dialog box to save Adapter settings. Once you click **OK** or **Apply** in the HP OpenView Adapter For ASAP Options dialog, all Adapter settings are saved.

## OVO Options

Once you open the HP OpenView Adapter For ASAP Options dialog box as described previously, click the OVO tab to configure OpenView interface-related options. These include settings that dictate how the Adapter updates OpenView service and node

information, how much detail is populated into the OpenView Services tree, message limit and acknowledgement options, and utility file locations:



VST023.vsd

## Service and Node Updates

The NonStop ASAP subsystem allows you to begin monitoring new objects at any time. These can be system objects such as CPUs and processes or application-defined objects monitored using the ASAPX API. As these objects are added, information on them is sent to the ASAP Client, which in turn passes the data on to the ASAP OpenView Adapter. When the Adapter receives this data, it determines if new services or nodes have been added. If they have, OpenView needs to be updated to include these items.

However, issues can arise when simply performing these updates as they occur, because the OpenView Console caches information from the OpenView server database. Thus new service and node information added to OpenView by the Adapter

might not immediately appear in active OpenView Consoles because those Consoles might be utilizing their cached data instead.

To prevent possible confusion, and to give you better control over when OpenView service and node updates occur, the Adapter provides three different options for updating OpenView data, listed in the Service/Node Updates group on the "OVO" tab:

## Update Service and Node Information Automatically

If you select this option, the ASAP OpenView Adapter automatically updates service and node information as needed whenever it detects changes. When the Adapter performs an update, a message appears on the server informing you to restart any active OpenView Consoles in order to retrieve the latest changes.

## Prompt When Service or Node Updates Are Required

If you select this option, the ASAP OpenView Adapter prompts you whenever it detects that service or node updates are necessary. If you choose to continue, the Adapter will add the new services, nodes, or both to OpenView. Then restart any active OpenView Consoles to ensure that you have the latest data. If you choose not to continue, the Adapter does not send the updates to OpenView. Instead, you can send them manually later as described in [Manually Updating or Reloading OpenView Data](#). This option is the default.

## Never Update Automatically; Manual Updates Only

If you select this option, the Adapter never automatically updates OpenView. Instead, you have to manually update OpenView as needed. See [Manually Updating or Reloading OpenView Data](#).

## Manually Updating or Reloading OpenView Data

In addition to determining how the Adapter should handle service and node updates, the Service/Node Updates group also allows you to manually update or reload OpenView data:

- **Manually Updating Data**

The Service/Node Updates group contains two status fields at the bottom of the group to inform you if service or node updates are available. If they are, and you want to update OpenView, click **Update Now**. The Adapter adds the new service and node information to OpenView. Once the operation is complete, restart any active OpenView Consoles to retrieve the latest data.

- **Reloading All Service and Node Data**

In some cases you might want to force the Adapter to resend all service and node information to OpenView; for example, if you inadvertently deleted some ASAP services from OpenView itself. To resend, click **Reload OVO**. The Adapter adds all current service and node information to OpenView. In addition, the Adapter

reissues an OpenView message for each object in an error state. Once the operation is complete, restart any OpenView Consoles to ensure that you have the latest data.

---

**Note.** OpenView service, node, and message updates can take a significant amount of time to complete when many objects or messages are added to OpenView. Some updates might require several minutes for all data to appear in the OpenView Console. Ordinarily, this situation occurs only when the ASAP Client and ASAP OpenView Adapter are initially started or after OpenView data is manually reloaded. In those instances, the Adapter is potentially supplying a large amount of data to OpenView. However, the updates might also take longer to complete if you have a major outage within your NonStop Kernel network (for example, an entire system is taken off-line, and so on).

---

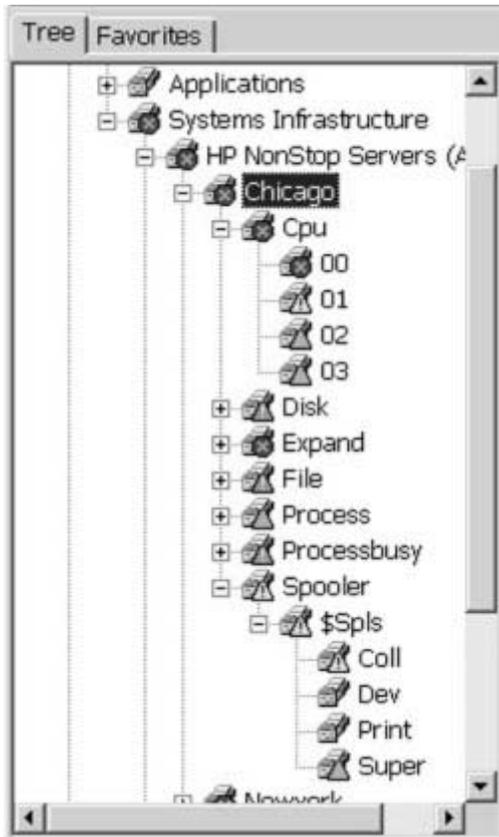
## Service Population Scheme

ASAP data is added to the OpenView Services tree whenever you begin monitoring a new set of NSK objects (for example, you start monitoring a new subsystem such as RDF or the Spooler, you add new application domains via ASAPX, or you add a new node to your ASAP configuration). All ASAP data in the Services tree view resides under HP NonStop Servers (ASAP) in the System Infrastructure category of services. The level of detail that the ASAP OpenView Adapter adds to this tree depends on the configured population scheme. Choices include populating all ASAP services, populating only those services that contain subservices, or populating to a fixed depth in the hierarchy. These options are in the Service Population Scheme group on the OVO tab:

- **Populate All Services**

If you choose to populate all services, all ASAP monitored objects are added to the OpenView Services tree view. For example, if ASAP is monitoring CPUs on system \Chicago, and that system contains four CPUs (00, 01, 02, and 03), the services Chicago, Chicago\CPU, Chicago\CPU\00, Chicago\CPU\01, Chicago\CPU\02, and

Chicago\CPU\03 are added to the Services tree view. The same strategy applies to all other entities and objects monitored by ASAP:



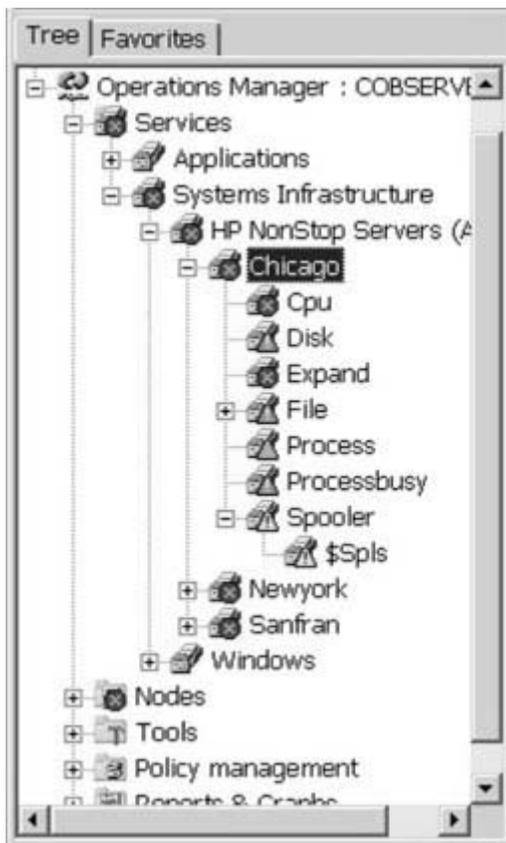
VST024.vsd

The advantage of populating all services is that it allows you to drill down to the NonStop Kernel object level to retrieve state information. This approach also greatly simplifies OpenView message filtering. As you drill down, you see only messages that pertain to the selected object. However, a cost is associated with this approach in terms of the number of services and objects added to OpenView. If you use ASAP to monitor a large network of systems and objects, you might not want to populate all services and objects.

- **Populate Container Services**

This option specifies that only services that contain subservices are added to the OpenView Services tree view. You see a tree view containing true categories of services, but the actual objects that make up those services are not present. The number of levels in the resulting OpenView hierarchy can vary from service to service. For example, if ASAP is monitoring CPUs on \Chicago, the Adapter adds the services Chicago and Chicago\CPU to OpenView. If ASAP is also monitoring the Spooler \$SPLS on \Chicago, the Adapter also adds the services Chicago\Spooler and Chicago\Spooler\ \$SPLS because the \$SPLS Spooler contains a set of subservices such as its supervisor, collector, and so on. In this

scenario, the Chicago\CPU hierarchy contains one level of detail fewer than the Chicago\Spooler hierarchy:



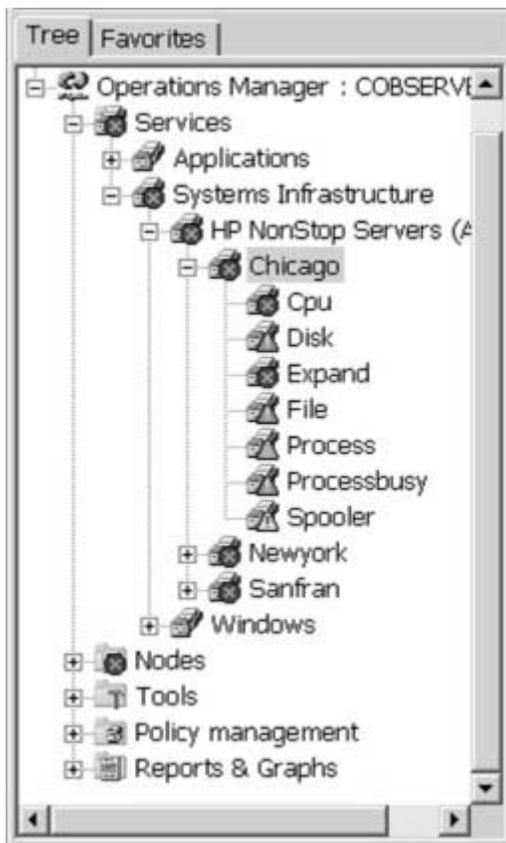
VST025.vsd

This approach provides a high level of detail on the state of various services, allowing you to drill down to determine exactly which services are in an error state. At the same time, because each object monitored by ASAP is not added to the tree view, the number of services in OpenView is minimized, making it much easier to manage a large network of systems. This option is the default.

- **Populate to  $n$  Levels of Detail**

If you want to more closely control the level of detail added to the OpenView Services tree view, you can opt to populate services to a fixed depth. The value you supply here defines how many levels of services are added to the Services tree view under HP NonStop Servers (ASAP). When the Adapter adds service information to OpenView, it stops adding levels of detail once this value is reached. The result is a balanced hierarchy within OpenView. For example, if ASAP is monitoring CPUs and the Spooler \$SPLS on system \Chicago, the Adapter adds

the services Chicago, Chicago\CPU, and Chicago\Spooler to OpenView if you have specified a depth of 2:



VST026.vsd

The Adapter can add levels of detail only if such data exists. Therefore, if you specify a depth of 4, for example, the Chicago\CPU hierarchy would still only extend 3 levels (Chicago\CPU\00, Chicago\CPU\01, and so on) because no objects are defined under the individual CPUs.

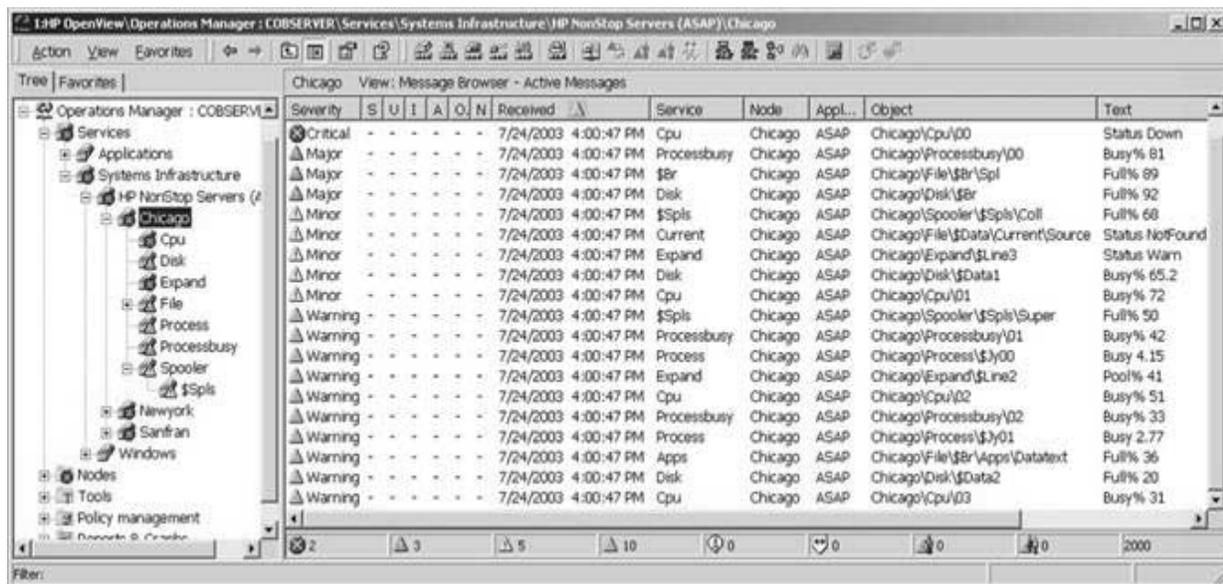
## The Services Tree and OpenView Messages

It is important to distinguish between the OpenView Services tree view and the messages logged to OpenView by the ASAP OpenView Adapter. The main purpose of the Services tree view is to categorize services and make it easier for you to determine, to whatever level you have configured, the state of the various services that make up your network of systems. The more levels of service detail you add to the tree view, the better you can determine where problems exist in your network and the more quickly you can view messages related to those services.

However, the content of the Services tree view has virtually no impact on the content of the messages that the Adapter logs to OpenView. The Adapter always logs messages against the actual low-level object that reported an error. The Object column in the OpenView Message Browser always contains the name of the specific object that caused that message to be generated, regardless of the level of detail in the Services

tree view. The only portion of message content that is influenced by the Services tree view is the service against which a given message is logged. This information appears in the Service column in the Message Browser.

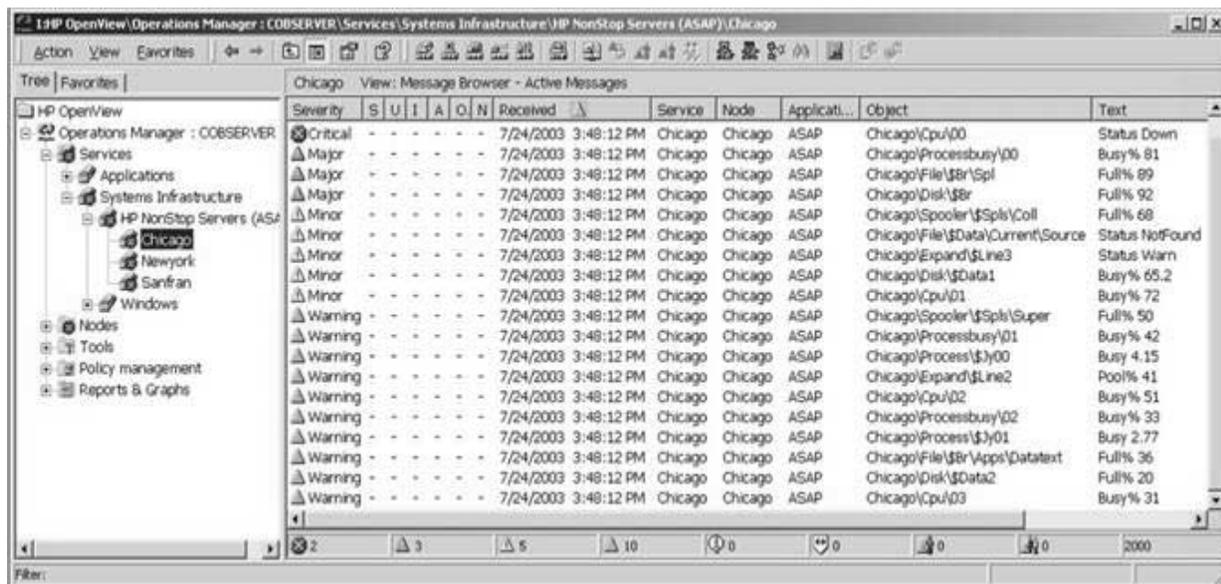
For example, assume that you have chosen to populate all container services. And assume that you are monitoring CPUs on system \Chicago and that CPU 00 is in a critical state. The ASAP OpenView Adapter logs a message with a severity of Critical to OpenView and specifies the object \Chicago\CPU\00. Because you populated the Services tree view with containers only, the message is logged against the service CPU on the node Chicago:



VST027.vsd

If instead you decide to populate the Services tree view to 1 level of detail (that is, systems only), the message described previously is the same except that it is logged

against the service Chicago, because no additional levels of services are present in the tree. The message is still generated for the object Chicago\CPU\00 as before:



VST028.vsd

## Deleting Services and Nodes From OpenView

The ASAP OpenView Adapter does not automatically remove any of the services or nodes that it previously added to OpenView. Once a service or node is added to the OpenView environment, messages can be logged against it, and that service or node becomes part of the OpenView configuration. Even if ASAP is no longer monitoring a given NonStop Kernel object, you can still view the history of messages logged against that object within OpenView.

If at some point your ASAP configuration changes and you want to delete one or more services or nodes from OpenView, you can use the OpenView Console. For example, suppose you alter the Service Population Scheme in a way that reduces the number of levels of service detail added (that is, you choose to now populate 1 level of detail, but previously you populated 2 levels). Removing a given service or node from the OpenView configuration also removes any messages associated with that object.

## Messaging Options

The ASAP OpenView Adapter logs a message whenever an ASAP-monitored object changes state. These messages appear in the OpenView Message Browser window. The details of each message include the name of the object, the service it is part of, the node it resides on, a severity level, and text describing the specific error or condition that caused the message.

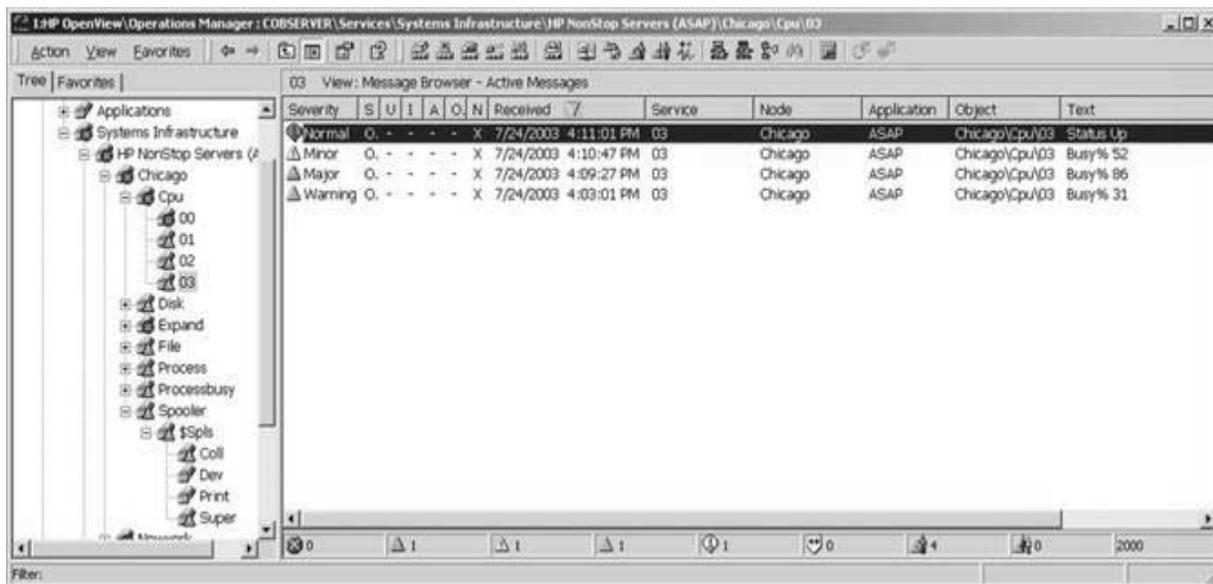
## Automatic Message Acknowledgement

The state of a given object can change many times while it is being monitored. Each of these state changes causes the Adapter to log a new message to OpenView. When OpenView receives a message for an object that already has a message logged against it, it can proceed in one of two ways:

- It can add the new message to the list of messages logged against the object.
- It can automatically acknowledge prior messages and display only the latest message in the active Message Browser.

To can configure this behavior, use the Acknowledge Messages Automatically check box under OVO Messages on the OVO tab. To acknowledge automatically and thereby display only the latest message for each object, select the check box. To allow messages to accumulate for each object, clear the check box. You can still acknowledge any of the messages manually via the OpenView Console.

The next example shows a sample of the contents of the Message Browser for the object \Chicago\CPU\03 with automatic acknowledgement disabled. The result is the history of messages for the object. In this scenario, the Services tree view reflects the state of the *most critical* unacknowledged message for the object, not the *latest* message:



VST029.vsd

Automatic acknowledgement is enabled by default.

## Limiting the Number of Messages Logged

In some cases (for example, if an entire NonStop system is being taken off the network), hundreds or thousands of ASAP-monitored objects can change state in an

instant. The number of messages logged by the ASAP OpenView Adapter as a result can have a significant impact on your OpenView server. To avoid this situation, you can regulate the number of messages generated by the Adapter in a given interval. To do so, select the Limit To *n* Msg / Minute check box under OVO Messages on the OVO tab, and supply the maximum number of messages that should be logged per minute.

If this option is enabled, the Adapter does not exceed the specified limit during any one minute interval. But the Adapter retains information on all monitored objects and can catch up, if needed. It continues to log the maximum number of messages per minute until messages are logged for all necessary objects. In addition, the Adapter always logs messages in order from most severe to least severe. The most critical messages are always sent to OpenView first.

This option is enabled by default, with a message limit of 100 messages per minute.

## Utility Files

As part of its normal operation, the ASAP OpenView Adapter requires access to the ovpmutil.exe and opcmsg.exe OpenView utility programs. They are used to add service and node data to OpenView and to route messages.

## File Locations

Typically these utilities are located in the HP OpenView program directory (or one of its subdirectories), and those directories are added to the Windows search path. In some cases, though, these files might reside in a different location, or the directories might not be on the Windows search path. If this situation occurs, you can specify the file locations manually. To do so, enter the full path and file name of the ovpmutil.exe file in the Ovpmutil Location field in the Utility group on the OVO tab, and enter the full path and file name of the opcmsg.exe file in the Opcmsg Location field. To locate and select the files interactively, click **Browse...** next to each of the respective fields.

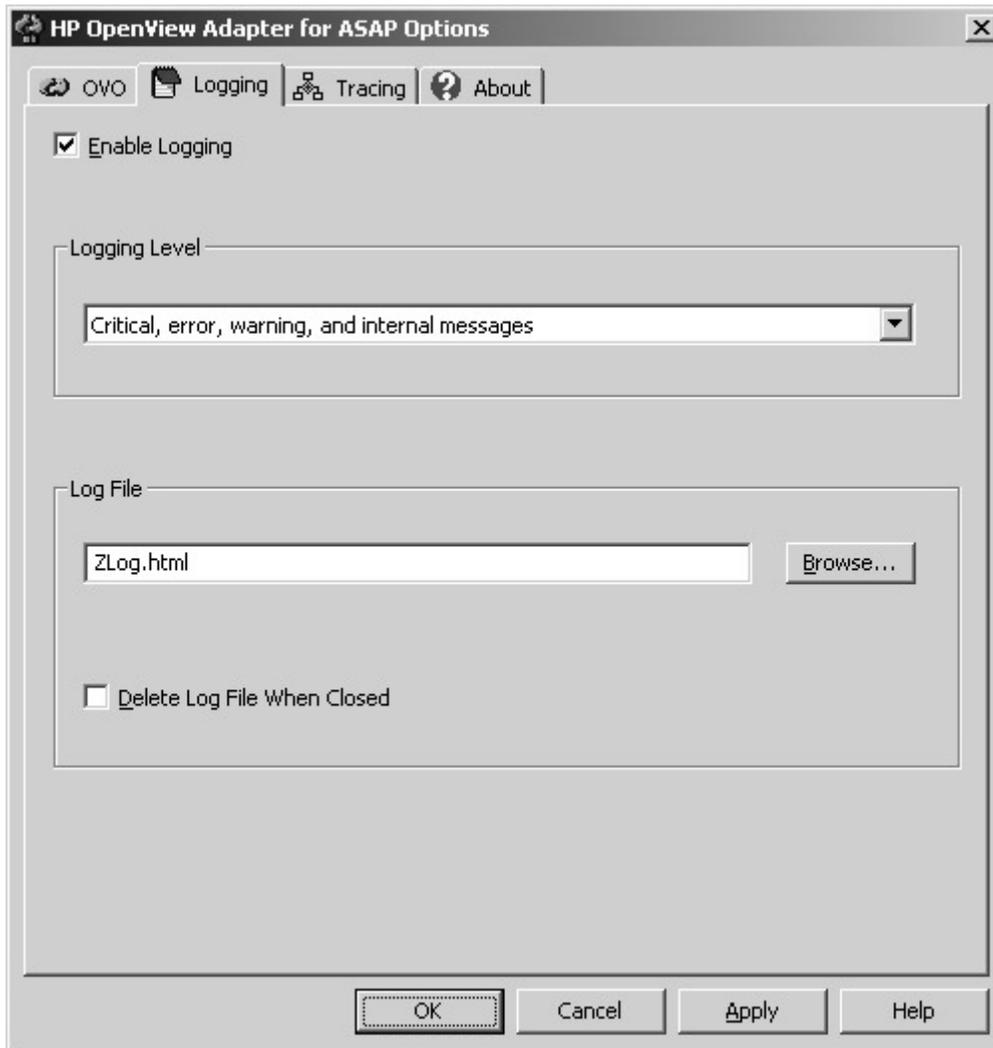
## Cleaning Up Temporary Files

In some cases, the utility programs generate temporary data files that are stored in the Adapter program directory. In general, there is no reason to retain these files, and the Adapter automatically deletes them as necessary. However, at times you might find it useful to keep them, particularly if you are attempting to resolve a problem. To specify whether the Adapter should automatically delete these temporary files, use the Cleanup Temporary Files Automatically check box in the Utility group on the OVO tab. If you enable this option, the Adapter automatically deletes any temporary files. If you disable this option, the temporary files remain in the Adapter program directory. This option is enabled by default.

## Logging Options

The ASAP OpenView Adapter provides a logging mechanism to report run-time errors and Adapter state information. You can define which types of information should be

logged (errors, warnings, and so on) and specify the log file name and location. To configure logging-related options, click the Logging tab in the HP OpenView Adapter For ASAP Options dialog box:



VST030.vsd

## Enabling and Disabling Logging

To enable the Adapter logging facility, select the Enable Logging check box. To disable logging, clear the check box. Logging is disabled by default.

## Logging Level

The logging level specifies which types of messages are logged by the Adapter:

Message Type	Description
Internal	Internal messages are generated from within the logging facility itself. If logging is enabled, internal messages are automatically enabled as well.
Critical	Critical messages are generated if the Adapter encounters a problem that it cannot recover from or that will prevent it from functioning correctly.
Error	Error messages are generated if the Adapter encounters a problem from which it can recover and continue to run, but potentially in degraded fashion.
Warning	Warning messages are generated if the Adapter encounters a problem from which it can recover and continue to run normally.
Info	Info messages are generated to provide status information on operations performed by the Adapter.

To specify which message types are logged, from the Logging Level menu on the Logging tab, select the appropriate option:

Internal messages only

Critical and internal messages

Critical, error, and internal messages

Critical, error, warning, and internal messages

All messages

## Log File Name

You can define the name and location of the log file used by the Adapter. By default, this file is Zlog.html, located in the same directory as the Adapter program file.

However, you can change the location, file name, or both by entering a new value in the Log File field on the Logging tab. To locate and select the file interactively, click **Browse...**

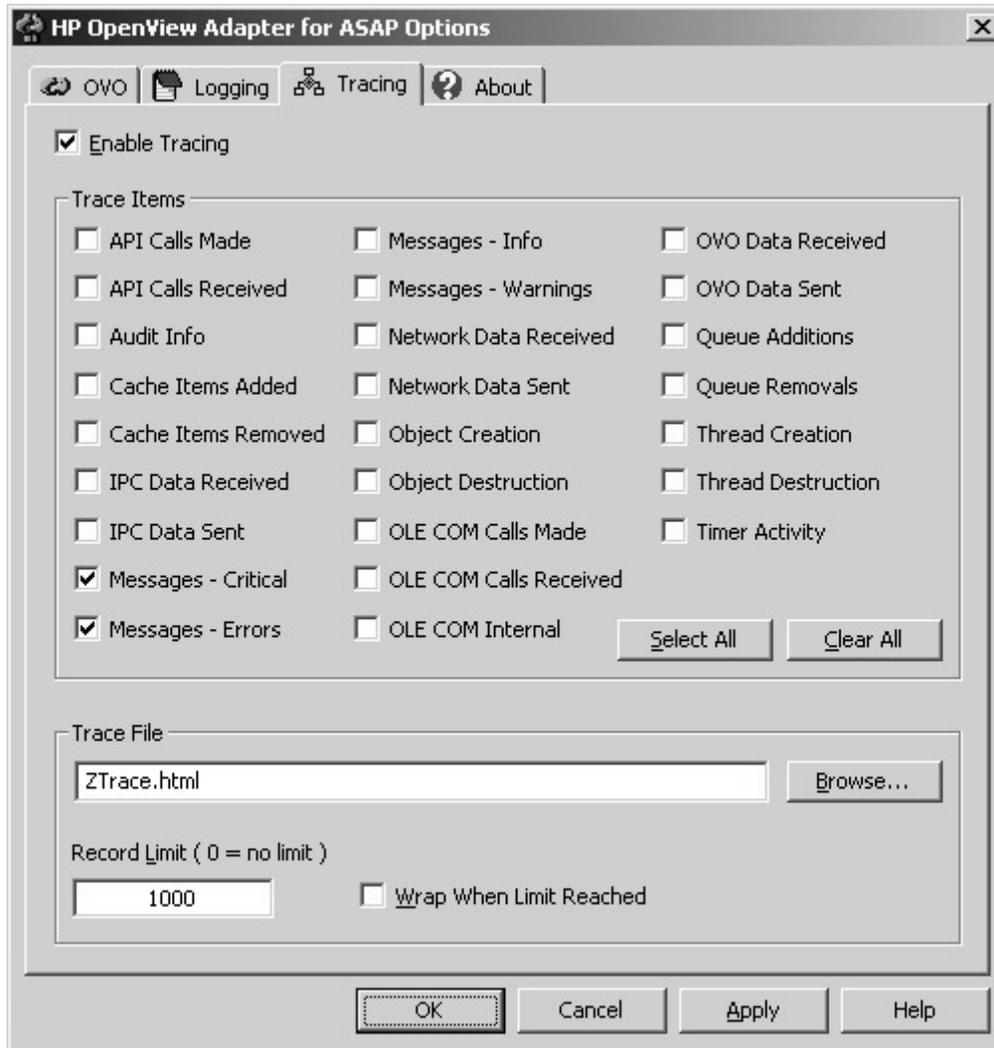
## Deleting the Log File

If you run with logging enabled, the log file can grow large over a long period of time. Although you can manually delete this file, the Adapter can also automatically delete the file whenever it is closed (for example, if you change the log file name or shut down the Adapter). This approach gives you access to the log data while the Adapter is running but helps prevent the log file itself from growing too large.

To automatically delete log files whenever they are closed, select the Delete Log File When Closed check box on the Logging tab. To retain all log data, clear the check box. By default, all data is retained.

# Tracing Options

The ASAP OpenView Adapter provides a tracing facility that can diagnose problems with the Adapter or with OpenView. You can define which types of information should be traced, specify the trace file name and location, and limit the amount of trace data collected. To configure tracing-related options, click the Tracing tab in the HP OpenView Adapter For ASAP Options dialog box:



VST031.vsd

## Enabling and Disabling Tracing

To enable the Adapter tracing facility, select the Enable Tracing check box. To disable tracing, clear the check box. Tracing is disabled by default.

Tracing can have a significant impact on the performance of the Adapter and should be enabled only if you are attempting to diagnose a problem. In general, activate tracing only if HP support staff ask you. If you enable tracing, attempt to have it enabled for the

minimum amount of time possible. Try to reproduce the problem scenario quickly, and then immediately disable tracing. Doing so has two advantages:

- It increases the likelihood that you will capture relevant trace information.
- It limits the impact on the Adapter and your operations management environment.

## Trace Items

The Adapter allows you to trace any combination of these types of data:

Trace Item	Description
API Calls Made	COM API calls made by the Adapter
API Calls Received	COM API calls received by the Adapter
Audit Info	Various Adapter internal statistics and operations
Cache Items Added	Activity related to adding items to the Adapter's object cache
Cache Items Removed	Activity related to removing items from the Adapter's object cache
IPC Data Received	Data sent to the Adapter from another process
IPC Data Sent	Data sent by the Adapter to another process
Messages - Critical	Critical errors that occur during normal processing and prevent the Adapter from operating correctly
Messages - Errors	Errors that occur during normal processing from which the Adapter can recover and continue to run, and potentially in degraded fashion
Messages - Info	Informational messages generated during normal processing
Messages - Warnings	Errors that occur during normal processing from which the Adapter can recover and continue to run normally
Network Data Received	Data received by the Adapter from a network client
Network Data Sent	Data sent by the Adapter to a network client
Object Creation	Internal objects created by the Adapter; tracing these items has a severe impact on Adapter performance and should only be enabled if you are diagnosing an object-related problem
Object Destruction	Internal objects destroyed by the Adapter; tracing these items has a severe impact on Adapter performance and should only be enabled if you are diagnosing an object-related problem
OLE COM Calls Made	OLE/COM calls made by the Adapter
OLE COM Calls Received	OLE/COM calls received by the Adapter
OLE COM Internal	OLE/COM internal activity within the Adapter
OVO Data Received	Data received from OpenView by the Adapter
OVO Data Sent	Data sent to OpenView by the Adapter
Queue Additions	Activity related to adding items to the Adapter's internal queues

Trace Item	Description
Queue Removals	Activity related to removing items from the Adapter's internal queues
Thread Creation	Activity related to starting new threads within the Adapter
Thread Destruction	Activity related to threads ending within the Adapter
Timer Activity	Timer-based processing and status information; tracing this item can have a severe impact on Adapter performance and should only be enabled if you are diagnosing a timing-related problem

To specify the items to trace, select the check box corresponding to the items you are interested in. To select all items, click **Select All**. To clear all items, click **Clear All**.

## Trace File Name

You can define the name and location of the trace file used by the Adapter. By default, this file is Ztrace.html, located in the same directory as the Adapter program file. However, you can change the location, file name, or both by entering a new value in the Trace File field on the Tracing tab. Or click **Browse...** to locate and select the file interactively.

No trace data is written to this file until tracing is disabled or the Adapter shuts down.

## Record Limit

Depending upon how long tracing is active and the number and types of trace items selected, trace data can accumulate rapidly or slowly. All trace data is collected in memory and is written to the trace file only when tracing is disabled or the Adapter shuts down.

To control how much trace data is collected, you can limit the number of trace records retained within the Adapter using the Record Limit field on the Tracing tab. Once this limit is reached, the Adapter stops collecting trace data unless wrapping is enabled. See [Wrapping Trace Data](#).

To specify a new record limit, enter the new value in the field. If you do not want the Adapter to impose a limit, enter 0 in this field. The default value is 1000 records.

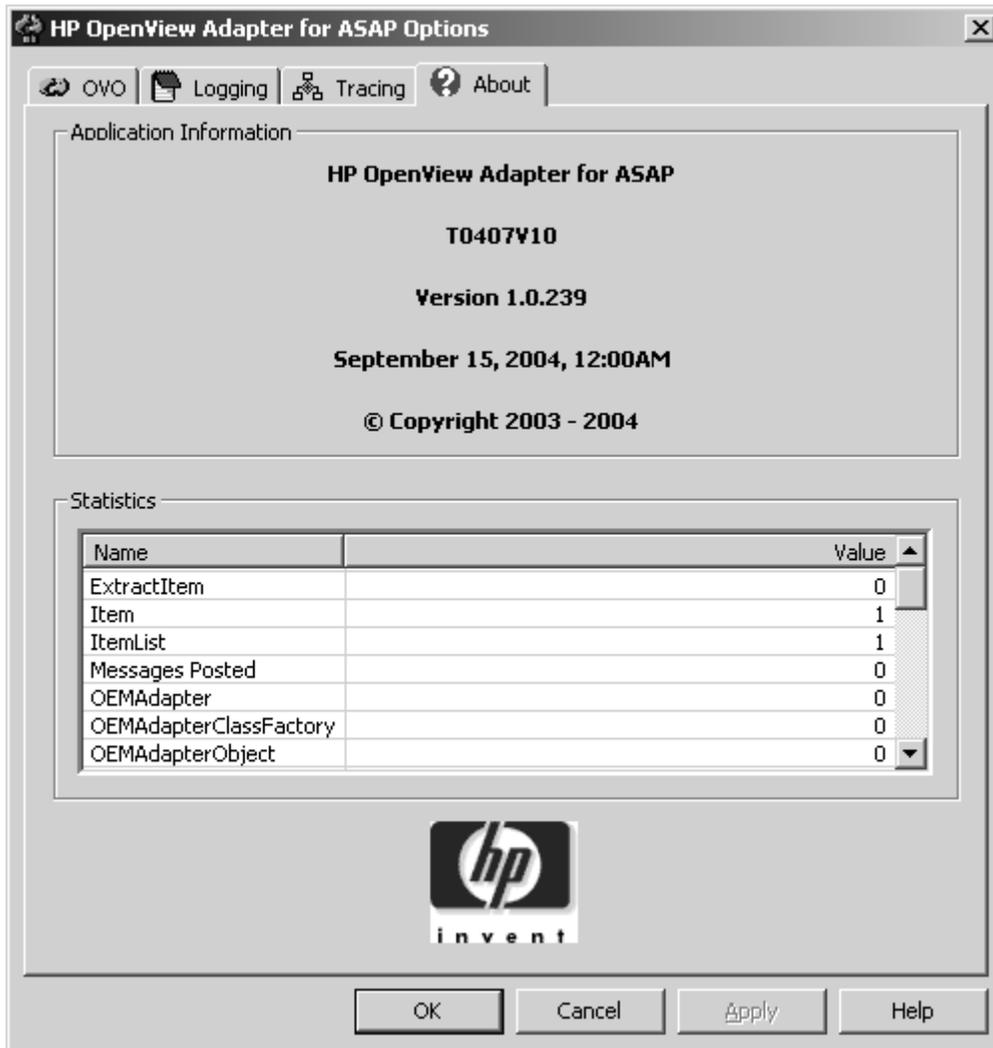
## Wrapping Trace Data

In most cases, you do not know ahead of time how many trace records are needed to capture all the data related to a given problem. Instead, you usually run and test, and eventually the error occurs. In most cases you do not want to collect an unlimited number of trace records, so the Adapter allows you to specify that trace data should automatically wrap whenever the record limit is reached. When the given record limit is reached, each subsequent new trace record replaces the oldest existing trace record. You always have the most recent trace data, but you can control how much data is retained by the Adapter.

To enable wrapping, select the Wrap When Limit Reached check box on the Tracing tab. To disable wrapping (and thereby retain the oldest trace data), clear the check box. This option is disabled by default.

## Version Information

Adapter version and statistics information are located on the About tab in the HP OpenView Adapter For ASAP Options dialog:



VST032.vsd

## OpenView Adapter Version Data

The Adapter version number and date appear under Application Information. If you need to report a problem with the Adapter, HP support staff will need this information.

## Statistics

Adapter internal statistics used for debugging appears under Statistics. HP support staff might need these statistics if a problem occurs with the Adapter.

## Command-Line Options

You can run the ASAP OpenView Adapter interactively with command-line options. These options cause the Adapter to perform a configuration-related function and then exit. In these cases, you cannot supply multiple command-line options, because the Adapter exits before processing any subsequent options.

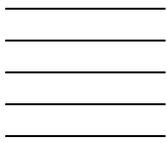
To run the Adapter with a command-line option:

```
OVOAdapter -option
```

Where *option* is one of:

Option	Description
CheckReg	Validates Adapter registration information and displays a confirmation dialog box listing any missing items. The Adapter exits immediately after processing this request.
CleanReg	Removes all Adapter user-configured options from the system registry but does not display a confirmation dialog box. The Adapter exits immediately after processing this request.
CleanRegStatus	Removes all Adapter user-configured options from the system registry and displays a confirmation dialog box. The Adapter exits immediately after processing this request.
Configure	Displays the Adapter's configuration options dialog box, allowing you to change any Adapter settings. Once you close the dialog box, the Adapter prompts you to continue normal execution or exit.
Help	Displays a list of all valid command-line options. The Adapter exits immediately after processing this request.
RegServer	Registers the Adapter but does not display a confirmation dialog box. The Adapter exits immediately after processing this request.
RegServerStatus	Registers the Adapter and displays a confirmation dialog box. The Adapter exits immediately after processing this request.

<b>Option</b>	<b>Description</b>
TraceMask <i>value</i>	Sets the Adapter's trace mask to the supplied value, where <i>value</i> is a hexadecimal string corresponding to the trace mask. This option is intended for debugging only. The Adapter continues normal execution after processing this request.
UnRegServer	Removes registration information for the Adapter but does not display a confirmation dialog box. The Adapter exits immediately after processing this request.
UnRegServerStatus	Removes registration information for the Adapter and displays a confirmation dialog box. The Adapter exits immediately after processing this request.



# Index



---

---

---

---

---

# Content Feedback

First Name: \_\_\_\_\_  
Phone: \_\_\_\_\_  
Company: \_\_\_\_\_

Last Name: \_\_\_\_\_  
e-mail address: \_\_\_\_\_

*(All contact information fields are required.)*

If you're reporting an error or omission, is your issue:

- Minor:** I can continue to work, but eventual resolution is requested.
- Major:** I can continue to work, but prompt resolution is requested.
- Critical:** I cannot continue to work without immediate response.

Comments (give sufficient detail to help us locate the text):

---

---

---

---

---

Thank you for taking the time to provide us with your comments.

You can submit this form online, e-mail it as an attachment to [pubs.comments@hp.com](mailto:pubs.comments@hp.com), fax it to 408-285-5520, or mail it to:

Hewlett-Packard Company  
NonStop Enterprise Division  
19333 Vallco Parkway, MS 4421  
Cupertino, CA 95014-2599  
Attn.: Product Manager, Software Publications



i n v e n t

<u>What's New in This Manual</u>	iii
<u>Manual Information</u>	iii
<u>New and Changed Information</u>	iii
<u>About This Manual</u>	v
<u>Notation Conventions</u>	v
<u>Abbreviations</u>	ix

## **1. Introduction to the OpenView Smart Plug-In for ASAP**

<u>Overview</u>	1-1
<u>ASAP Data Within OpenView</u>	1-2
<u>The OpenView Services Tree</u>	1-2
<u>The OpenView Nodes Tree</u>	1-3
<u>The OpenView Message Browser</u>	1-5
<u>OpenView and OEM State Models</u>	1-7

## **2. Installing the OpenView Smart Plug-In for ASAP**

<u>System Requirements</u>	2-1
<u>Hardware Requirements</u>	2-1
<u>Software Requirements</u>	2-1
<u>Installation Procedure</u>	2-1
<u>Installation - Automated</u>	2-2
<u>Installation - Manual</u>	2-2
<u>Verifying the Installation</u>	2-5
<u>Files Installed</u>	2-7

## **3. Removing the OpenView Smart Plug-In for ASAP**

<u>Policy Removal Procedure</u>	3-1
<u>Adapter Removal Procedure</u>	3-2
<u>Removal - Automated</u>	3-2
<u>Removal - Manual</u>	3-3
<u>Files Removed</u>	3-4

## **4. Configuring the ASAP OpenView Adapter**

<u>OEM Gateway Adapter Configuration</u>	4-1
<u>Accessing the ASAP OpenView Adapter Configuration Dialog</u>	4-4
<u>OVO Options</u>	4-6
<u>Service and Node Updates</u>	4-7
<u>Manually Updating or Reloading OpenView Data</u>	4-8
<u>Service Population Scheme</u>	4-9
<u>Deleting Services and Nodes From OpenView</u>	4-14

<u>Messaging Options</u>	4-14
<u>Utility Files</u>	4-16
<u>Logging Options</u>	4-16
<u>Enabling and Disabling Logging</u>	4-17
<u>Logging Level</u>	4-18
<u>Log File Name</u>	4-18
<u>Deleting the Log File</u>	4-18
<u>Tracing Options</u>	4-19
<u>Enabling and Disabling Tracing</u>	4-19
<u>Trace Items</u>	4-20
<u>Trace File Name</u>	4-21
<u>Record Limit</u>	4-21
<u>Wrapping Trace Data</u>	4-21
<u>Version Information</u>	4-22
<u>OpenView Adapter Version Data</u>	4-22
<u>Statistics</u>	4-23
<u>Command-Line Options</u>	4-23

## Index

# Examples



## Figures

Figure 1-1. OpenView Smart Plug-In for ASAP architecture 1-2



# Tables