

HP OpenView Performance Agent

For the Linux Operating System

Software Version: 4.60

Installation and Configuration Guide

Manufacturing Part Number: B7491-90080

Document Release Date: December 2006

Software Release Date: December 2006



Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

© Copyright 1983-2006 Hewlett-Packard Development Company, L.P.

Trademark Notices

UNIX® is a registered trademark of The Open Group.

Adobe®, Acrobat®, and PostScript® are trademarks of Adobe Systems Incorporated.

Motif® is a registered trademark of the Open Software Foundation in the U.S. and other countries.

All other product names are the property of their respective trademark or service mark holders and are hereby acknowledged.

Support

You can visit the HP OpenView Support web site at:

www.hp.com/managementsoftware/support

HP OpenView online support provides an efficient way to access interactive technical support tools. As a valued support customer, you can benefit by using the support site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract.

To find more information about access levels, go to:

www.hp.com/managementsoftware/access_level

To register for an HP Passport ID, go to:

www.managementsoftware.hp.com/passport-registration.html

Contents

1	Installing or Upgrading OpenView Performance Agent	7
	Introducing OpenView Performance Agent	7
	Installation Requirements	9
	Supported Architecture and Operating System Versions	9
	Disk Space	9
	Install or Upgrade Procedures	11
	Upgrade	11
	First time Installation	11
	Stopping Active Performance Tool Processes	11
	Installing OV Performance Agent	12
	Installing OV Performance Agent on Debian	14
	Installing OV Performance Agent Using OV Operations	14
	OV Performance Agent on a Virtualized Environment	15
	OV Performance Agent on the Service Console	15
	OV Performance Agent on a Virtual Machine	15
	Removing OV Performance Agent	17
2	Starting and Running OV Performance Agent	19
	Starting and Stopping OV Performance Agent	20
	Starting and Stopping Automatically	21
	Status Checking	22
	Examples Directory	22
	Communicating Across a Firewall	23
	Configure OVPA Ports	25
	Configure HTTP Clients in a Firewall Environment	27
	Verify Firewall Configuration	29
	Configuring Systems with Multiple IP Addresses	29
	Configuring Secure Communication	30

Using Certificates	30
Using Client Authentication	30
Enabling Authentication with the authip File.	31
Formatting the authip File	31
Configuring Data Sources	33
Datasources Configuration File Format	33
Deleting Data Sources	34
Parm File	35
Defining Alarms	36
3 Documentation	39
OV Performance Agent Documentation Set	40
Viewing Documents on the Web	41
Adobe Acrobat Files	42
A Configuring Coda	43
Introduction	43
Glossary	47
Index	55

1 Installing or Upgrading OpenView Performance Agent

Introducing OpenView Performance Agent

HP OpenView Performance Agent (OVPA) captures performance, resource, and transaction data from your Linux system. Using minimal system resources, the software continuously collects, summarizes, time stamps, and detects alarm conditions in current and historical resource data across your system.

OV Performance is now supported on VMware ESX Server. For more information, refer to the section titled [OV Performance Agent on a Virtualized Environment](#) on page 15.

You can analyze the data using local export capabilities, or complementary OpenView products such as OV Reporter and OV Performance Manager (OVPM) 4.0 and later. Also, OV Performance Agent provides data access and sends alarm notifications to OpenView Network Node Manager (NNM) and OpenView Operations (OVO).



OV Performance Manager (OVPM) in this document refers only to version 4.0 and later for UNIX and Windows platforms. OVPM 3.x (PerfView) does not connect to OV Performance Agent for Linux.

OV Performance Agent uses Data Source Integration (DSI) technology to receive, alarm on, and log data from external data sources such as applications, databases, networks, and other operating systems.

The comprehensive system performance data recorded and stored by OVPA from your system allows you to perform the following tasks:

- Characterize the workloads in the environment.
- Analyze resource usage and load balance.
- Perform trend analyses on historical data to isolate and identify bottlenecks.

- Respond to error conditions.
- Perform service-level management based on transaction response time.
- Perform capacity planning.
- Solve system management problems before they arise.

For a comprehensive description of OV Performance Agent, see the *HP OpenView Performance Agent for UNIX User's Manual*.

Installation Requirements

Before installing OV Performance Agent, make sure that your system meets the requirements described in this section.

Supported Architecture and Operating System Versions

The table below lists the supported architecture and operating system versions, as well as the data communication protocols supported by each combination.

Architecture	Operating System
On 32-bit Intel Pentium (x86)	<ul style="list-style-type: none">• RedHat Enterprise Linux (ES/AS/WS) 2.1, 3 and 4• SuSE Enterprise Server 8.x, 9.x and 10.x• Debian 3.0 r2/r3/r4, 3.1• TurboLinux Server 8 and 10• Asianux 2.0 (Miracle Linux)
On 64-bit systems (x86_64)	<ul style="list-style-type: none">• RedHat Enterprise Linux (ES/AS/WS) 3 and 4• SuSE Enterprise Server 9.x and 10.x• Asianux 2.0 (Miracle Linux)

Disk Space

OV Performance Agent installs in the `/opt/perf/` and `/opt/OV/` directories and creates its log and status files in the `/var/opt/perf/` and `/var/opt/OV/` directories.

- For first time installation of OV Performance Agent, approximately 50 MB of disk space is required in the `/opt/perf/` and `/opt/OV/` directories.
- For OVPA databases and status files, allow for 125 MB of disk space in the `/var/opt/perf/` and `/var/opt/OV/` directories.

When OV Performance Agent is first installed, the default `parm` file is set up to collect a maximum of approximately 60 MB of log file data.

You can modify the `parm` file and restart `scopeux` to change the size of the log files. For information about modifying the `parm` file and how it is used to limit and configure log file data storage, see Chapter 2 of the *HP OpenView Performance Agent for UNIX User's Manual*.

Install or Upgrade Procedures

OV Performance Agent is on CD-ROM installation media. The size of the product is approximately 50 MB, including the product documentation.

Upgrade

If you have previously installed OVPA or GlancePlus on the system, you must perform the following tasks:

- Stop any performance tools or processes that may be running. For instructions, refer to the section [Stopping Active Performance Tool Processes](#).
- Install OVPA. For instructions to install OpenView Performance Agent, refer to the section, [Installing OV Performance Agent](#).

First time Installation

If you are installing OV Performance Agent for the first time, for instructions, refer to the section, [Installing OV Performance Agent](#).

Stopping Active Performance Tool Processes

- 1 Log in as **root**.
- 2 Run `perfstat` to check for active performance tools by typing:
`/opt/perf/bin/perfstat`
If `perfstat` reports any active performance tool processes, stop them. (Make sure that you have exited these tools before doing so.)
- 3 If a previously installed version of OV Performance Agent is running, stop it by typing:

```
/opt/perf/bin/ovpa stop
```

▶ Customized configuration files such as the `parm`, `alarmdef`, `ttd.conf` and `datasources` as well as any customized log files will not be overwritten by the new installation. The new configuration files are installed in the `/opt/perf/newconfig/` directory.

- 4 Make sure the `midaemon` (measurement interface daemon) is not active by typing:

```
ps -ef | grep midaemon
```

If the `midaemon` is still active, type:

```
/opt/perf/bin/midaemon -T
```

- 5 Make sure the `ttd` (transaction daemon) is not active by typing:

```
ps -ef | grep ttd
```

If the `ttd` is still active, type:

```
/opt/perf/bin/ttd -k
```

▶ If you stop `ttd`, you must stop all ARM-instrumented applications that are running before you restart `ttd` and OV Performance Agent processes.

- 6 Run `perfstat` again to ensure that no performance tools or processes are active. When all tools or processes have been stopped and your customized configuration files are backed up, proceed with the installation.

Installing OV Performance Agent

▶ If you are installing on Debian, please see [Installing OV Performance Agent on Debian](#) on page 14.

- 1 Make sure you are logged in as **root**.
- 2 Insert your installation CDROM into the drive.

Typically, Linux systems are configured to automatically mount the CD-ROM. If your system is not configured to automount the CDROM, then you must manually mount it in the File Manager, or enter the command:

```
mount -t <type> <device> <dir>
```

where *<type>* is the file system type, *<device>* is the CDROM device and *<dir>* is where the CD-ROM directory will be mounted. For example:

```
mount -t iso9660 /dev/cdrom /mnt/cdrom
```

- 3 Open a terminal window. Run the `ovpa.install` script from the command line.
- 4 In the terminal window, change your present working directory by typing:

```
cd /<directory>
```

where *<directory>* is your CDROM directory.

- 5 Type **ls** to verify that you are in the correct directory. You will see the README file, the `ovpa.install` and `glance.install` scripts, and the `paperdocs`, `packages`, and `ReleaseNotes` subdirectories.
- 6 Run the installation script by typing:

```
./ovpa.install
```

The `ovpa.install` script includes command line options for more advanced installations. The syntax of the command is:

```
ovpa.install [-n] [-f] [-v]
```

The command line options have the following meaning:

<code>-n</code>	Do not prompt for <code>ovpa</code> start at the end of installation
<code>-f</code>	Force installation (<code>--force</code> , <code>--nodeps</code>)
<code>-v</code>	Verbose installation

OV Performance Agent installation is now complete.

By default, the OV Performance Agent processes are configured to start automatically upon reboot. See [Starting and Stopping OV Performance Agent](#) on page 20.

Chapter 2, “Starting Up and Running OV Performance Agent”, describes tasks you may wish to perform to configure OV Performance Agent to support your environment.



For more information and details about new features and changes in OV Performance Agent, see the Release Notes, in `/opt/perf/ReleaseNotes/OVPA`.

Installing OV Performance Agent on Debian

On Debian, you must first install `/usr/bin/rpm` (RedHat Package Manager) if it is not already installed on your system. OV Performance Agent requires `rpm` for the installation.

If `rpm` has not been used previously, follow these steps:

- 1 Make sure you are logged in as **root**.
- 2 Create the `rpm` database by typing:

```
# mkdir /var/lib/rpm
# rpm --initdb --dbpath /var/lib/rpm
```
- 3 Follow Steps 2 through 6 in the section [Installing OV Performance Agent](#) on page 12.

Installing OV Performance Agent Using OV Operations

If you are using OV Operations for UNIX 8.x or later, you can install OV Performance Agent from the management server to a Linux managed node.

For installation instructions on OV Operations for UNIX versions 8.x, please see Chapter 4, “HP OpenView Performance Agent,” in the *HP OpenView Operations Administrator's Reference*.

OV Performance Agent on a Virtualized Environment

This version of OV Performance Agent is supported on VMware ESX Server. OV Performance Agent should be installed on service console of VMWare ESX Server. For more information, refer to the section [OV Performance Agent on the Service Console](#).

This version of OV Performance Agent also supports monitoring of Virtual Machines (VMs) hosted on VMWare ESX Server 2.5.2 and later. For more information, refer to the section [OV Performance Agent on a Virtual Machine](#).

OV Performance Agent on the Service Console

OV Performance Agent should be installed on the service console of a VMWare ESX Server. For installation procedures, refer to the section [Install or Upgrade Procedures](#). OV Performance Agent captures the following:

- System wide performance and physical system resource usage data.
- Individual Guest's performance and physical system resource usage data.

A new class of metrics, BYLS, is introduced to record performance and resource data for individual guest operating systems.




For complete list of metrics and descriptions, refer to the *Dictionary of Performance Metrics*.

OV Performance Agent on a Virtual Machine

This version of OVPA supports monitoring of Virtual Machines (VMs) hosted on VMWare ESX Server 2.5.2 and later. For installation procedures refer to the section, [Install or Upgrade Procedures](#).

Install **VMWare Tools** on the VM. OVPA installed on a VM, hosted on VMWare ESX Server 2.5.2 and later, requires VMWare tools to recognize the virtual machine as a Guest operating system.

On VMs hosted on VMWare ESX Server 3.0.1, enable the **VMware Guest API** on the VMs. To capture the Virtual Machine's utilization of physical resources of a system, OVPA uses the **VMWare Guest API**. For more information about **VMware Guest API**, refer to latest documentation available on **VMware Guest SDK** from VMWare.

 For complete list of metrics and descriptions, refer to the *Release Notes* and *Dictionary of Performance Metrics*.

The following table lists all the features which are supported and not supported in a virtualized environment:

Features/ Functionalities	Linux	VMware ESX Server
Alarming	Available	Available
Data Source Integration capabilities	Available	Available
Application metric class	Available	Not available
BYLS metric	Not available	Available
proc metric	Available	Not available
Transaction metric class	Available	Not available
Table class	Available	Not available
ARM	Available	Not available

Removing OV Performance Agent

If you want to remove OV Performance Agent from a system, run the `ovpa.remove` script available in the `/opt/perf/bin/` directory. Before removing OV Performance Agent, this script will check to see if you want to remove log and data files that were created.

The script supports the following command line options for a more flexible product removal. The syntax of the command is as follows:

ovpa.remove [-f] [-h] [-r]

-f Force removal. Do not prompt for confirmation before removing the product (non-interactive mode).

NOTE: This option will also remove GlancePlus, if it is installed.

-h Help: Displays this usage message.

-r Remove the OVPA configuration, data, and product files.

NOTE: Combining the `f` and `r` options does not prompt for confirmation and removes all product, configuration, and data files including Glance, if it is installed.

Some product packages may remain installed on the system, if those packages are shared across other OpenView products and are required by other tools. Such packages will be removed only when the last tool requiring them is also removed.

2 Starting and Running OV Performance Agent

This chapter describes the tasks involved in starting up and running OV Performance Agent after it has been installed on your system. The chapter covers the following:

- Starting and Stopping OV Performance Agent
- Communicating Across a Firewall
- Configure OVPA Ports
- Configuring Secure Communication
- Configuring Data Sources
- Defining Alarms



If you are planning to log data from other sources using data source integration (DSI), read the *HP OpenView Performance Agent for UNIX Data Source Integration Guide*.

Starting and Stopping OV Performance Agent

When installation is complete, OV Performance Agent can be started. If you ever need to stop, start, or restart OV Performance Agent, the `ovpa` script lets you stop or restart currently running processes.

To start OV Performance Agent and its processes:

- 1 Log in as **root**.
- 2 Type: **`/opt/perf/bin/ovpa start`**

The `ovpa start` script starts OV Performance Agent and all its processes, including the `scopeux` (data collector), `midaemon` (measurement interface daemon), `ttd` (transaction tracking daemon), `coda`, `ovc`, `ovbbccb`, and the alarm generator. As the script executes, the status of the processes that are started is displayed on the screen.

You can stop OV Performance Agent processes while they are running and restart them using the `ovpa` script and its appropriate options.

- `ovpa stop` stops all OV Performance Agent processes except `ttd` (the transaction tracking daemon), `ovc`, and `ovbbccb`. These processes must always be running. If OpenView Operations agent is running on the system, `ovpa stop` does not stop the `coda` daemon.



If you must stop `ttd`, any ARM-instrumented applications that are running must also be stopped before you restart `ttd` and OV Performance Agent processes.

- `ovpa restart server` causes `coda` to stop and then start, temporarily disabling alarming and access for clients such as OV Performance Manager, and reread the `datasources` file. It also stops and then starts the `perfalarm` processes and rereads the `alarmdef` file.
- `ovpa restart` causes `scopeux` and the server processes to temporarily stop and then start. It reads the `parm` file as well as forces the transaction daemon `ttd` to reread its configuration file `ttd.conf`.
- `ovpa restart alarm` causes the alarm generator process to reread the `alarmdef` file so that if you have made changes to the file, the new alarm definitions will take effect without restarting all OVPA processes. This action does not disrupt other processes.

Starting and Stopping Automatically

The process of automatically starting OV Performance Agent whenever the system restarts, and of stopping whenever the system shuts down is controlled by the OVPA system startup and shutdown script and the OVPA startup configuration file, which varies according to the specific requirements of your Linux distribution as follows:

Table 1 OVPA Startup Scripts and Configuration Files

Linux Distribution	OVPA Startup and Shutdown Script	OVPA Startup Configuration File
Red Hat	/etc/rc.d/init.d/ovpa	/etc/sysconfig/ovpa
SuSE	/etc/init.d/ovpa	/etc/ovpa.conf
TurboLinux	/etc/rc.d/init.d/ovpa	/etc/sysconfig/ovpa
Debian	/etc/init.d/ovpa	/etc/ovpa.conf
ESX Server	/etc/rc.d/init.d/ovpa	/etc/sysconfig/ovpa

The OVPA startup configuration file contains various environment variables that control the behavior of OV Performance Agent at system startup. This file is a source file for the OVPA system startup and shutdown script.

You can modify the following environment and shell variables to change the default behavior of OV Performance Agent:

- `OVPA_START` controls the auto-start of OV Performance Agent whenever your system restarts. The variable can have one of the following values:
 - 0 Do not start OV Performance Agent at system startup
 - 1 Start OV Performance Agent at system startup
- The `OVPA_START_COMMAND` is a variable that specifies the script options that are used to start OV Performance Agent whenever your system restarts. Normally, the variable is set to:

```
/opt/perf/bin/ovpa start
```



The OVPA startup configuration file is not overwritten when OV Performance Agent is patched or updated, so that any customizations you made are preserved.

Status Checking

Several status files are created in the `/var/opt/perf/` and `/var/opt/OV/` directories when OV Performance Agent is started. You can check the status of all or some OV Performance Agent processes using the `perfstat` command.

The following status files contain diagnostic information you can use to troubleshoot problems that may arise with the OV Performance Agent processes.

```
/var/opt/perf/status.scope  
/var/opt/perf/status.perfalarm  
/var/opt/perf/status.ttd  
/var/opt/perf/status.mi  
/var/opt/OV/log/coda.txt
```

Examples Directory

The `/opt/perf/examples/` directory contains examples of configuration, syntax, and sample program files that can be used to customize your HP Performance Tools. For example, the `/opt/perf/example/config/` subdirectory contains sample alarm definitions and examples of `parm` file application-specific parameters. For more information, see the `/opt/perf/examples/README` file.

Communicating Across a Firewall

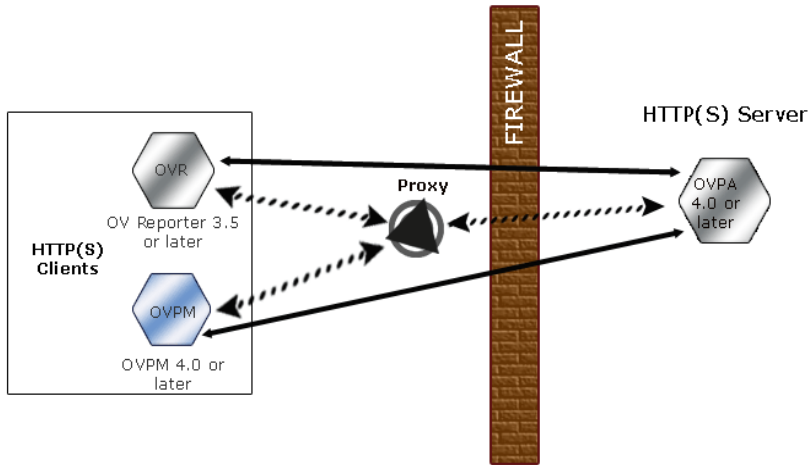
A firewall can be defined as a method for filtering the flow of data between one network and another. OV Performance Agent (OVPA) now supports HTTP based communications interface for data access between client and server applications, in addition to the previously supported communication mechanism through a packet-filtering network firewall. OVPA internally uses the Black Box Communication (BBC) component to implement HTTP based data communication. The BBC component provides a firewall-friendly single-port communication infrastructure based on HTTP or HTTPS (BBC5). The HTTP based interface is flexible, as it can use proxies, requires fewer ports, and is firewall friendly.

If a network firewall exists between two systems, OV Performance Manager and OV Reporter can get data from the OV Performance Agent system.



The name OV Performance Manager (OVPM) is used throughout this document to refer to OVPM versions 4.0 and later. OVPM 3.x does not connect to OVPA for Linux systems.

Figure 1 Communicating with OVPA in a Firewall Environment



For firewall configuration it is important to know which system initiates the communication (client) and which receives communication requests (server), so that the firewall rules can be set up accordingly. In a typical remote communication, a client, using the source port, connects to a server that is listening on the destination port on a remote system. Understand your firewall environment including the client and server data flow. To configure communications with OVPA in a firewall environment, perform the following tasks:

- 1 [Configure OVPA Ports.](#)
- 2 [Configure HTTP Clients in a Firewall Environment.](#)
- 3 [Verify Firewall Configuration.](#)

Figure 1 on page 24 shows how OVPA communicates with Reporter (version 3.5 or later) and OV Performance Manager (OVPM version 4.0 or later) through a firewall. OVPA is an HTTP or HTTPS server. Reporter and OVPM 4.x are HTTP clients. OVPM 5.0 can be an HTTP or HTTPS client. If an HTTP proxy is used, Reporter and OVPM communicate with OVPA via the proxy.



For Linux users, if you cannot upgrade OVPM 3.x to OVPM 4.0 or later, you must access the OVPA for Linux data locally via the `extract` program's `export` function.

Configure OVPA Ports

You can configure OVPA ports in a firewall environment in one of the two ways:

- [Configure Two-Port Communication](#)
- [Configure Single-Port Communication](#)
- [Verify Port Settings](#)

On an OVPA system using BBC5, by default, the BBC communication broker uses port 383 and `coda` uses a dynamically allocated port.

Configure Port Settings for the BBC Communication Broker

You can configure the port settings of the default port used by the BBC communication broker. Use the `ovconfchg` command to change the port settings on the OV Performance Agent system. You can use one of the following options:

```
— ovconfchg -ns bbc.cb.ports -set PORTS <host>:<port>
```

(Or)

```
— ovconfchg -ns bbc.cb.ports -set PORTS <domain>:<port>
```

```
Example: ovconfchg -ns bbc.cb.ports -set PORTS  
xyz.abc.com:50383
```

Restart `ovpa` using the following command:

```
ovpa restart server
```

Configure Two-Port Communication

By default, `coda` daemon uses a dynamically chosen second port, in addition to port 383 used by the BBC communication broker. You can configure the port settings of `coda` to listen at a well known port of your choice using the `ovconfchg` command. Type the following commands:

```
ovconfchg -ns coda.comm -set SERVER_PORT <portnumber>
```

```
ovpa restart server
```



Using a dynamic port when connecting to OVPA remotely through a firewall can be difficult, because you may not know the firewall ports to open.

Configure Single-Port Communication

On the OV Performance Agent system, the BBC communication broker uses port 383 and `coda` uses a port that is dynamically allocated. You can configure the port settings for `coda` to share the same port used by the communication broker using the `ovconfchg` command. You can use one of the following options:

```
ovconfchg -ns coda.com -set SERVER_BIND_ADDR localhost
```

```
ovpa restart server
```

Verify Port Settings

To verify the port settings, type the following command:

```
perfstat -d
```

The output displays the following information:

- port number of the port used by Coda
- port number of the port used by BBC communication broker
- the port settings configured
- indicates if secure communication is enabled
- indicated if `coda` metric collection is enabled

For example:

```
Datacomm configuration :  
-----
```

Coda Port	49552 (Dynamic)
	Two port Communication
BBC communication broker port	383
SSL security	NONE
Coda Metric Collection(Prospector)	Disabled

Configure HTTP Clients in a Firewall Environment

There are two ways to configure HTTP clients in a firewall environment:

- [Configuring HTTP Clients \(Reporter/OVPM\) with HTTP Proxy](#)
- [Configuring HTTP Clients \(Reporter/OVPM\) without HTTP Proxy](#)

In both cases, to access data from OVPA nodes, only one port needs to be opened on the HTTP server (OVPA) side.

Configuring HTTP Clients (Reporter/OVPM) with HTTP Proxy

It is recommended that you use HTTP proxies when communicating through a firewall. This simplifies the configuration by using proxies that are often already in use in your environment. The firewall must be open for exactly one port if proxies are to be used in both directions. To access data collected by OVPA, ports for the HTTP server (OVPA) and the HTTP client (Reporter and OVPM) must be opened.



It is recommended that you do not change the default 383 port.

When an HTTP proxy is used, Reporter and/or OVPM for Windows and UNIX need to be configured to specify the proxy to be used to contact OVPA.

To configure OVPM versions 5.0 and later, and Reporter 3.7:

Type the following command,

```
ovconfchg -ns bbc.http -set PROXY proxy:port+(a)-(b)
```

The variables *a* and *b* are comma separated lists of hostnames, networks, and IP addresses that apply to the proxy. Multiple proxies may be defined for one PROXY key using the “;” or “,” delimiter. “-” before the list indicates that those entities do not use this proxy, “+” before the list indicates that those entities do use this proxy. The first matching proxy is used.

To configure Reporter versions 3.6 and earlier, and OVPM 4.x:

Edit the `/var/opt/OV/conf/BBC/default.txt` configuration file.

In the [DEFAULT] section of the `default.txt` file, locate the lines that relate to the PROXY and set the PROXY parameter as follows.

```
PROXY web-proxy.hp.com:8088-(localhost, *.hp.com) + (*)
```

In this example, the proxy `web-proxy` will be used with port 8088 for every server (*) except requests for the local machine (`localhost`) and requests internal to HP (matching `*.hp.com`, for example **`www.hp.com`**).

Configuring HTTP Clients (Reporter/OVPM) without HTTP Proxy

If HTTP proxies are not available, additional configuration settings are required on the HTTP clients (Reporter and OVPM system).

If Reporter and OVPM for Windows are installed on the same system and both access OVPA in parallel, you can specify a port range as described in this section. If they are running on different systems, you can specify a single port for each. Depending on the versions of OVPM and Reporter you are using select from the following options:

Configure OVPM 5.0 and later, and Reporter 3.7 as follows:

Type the following command,

```
ovconfchg -ns bbc.http -set CLIENT_PORT <port range>
```

Where *<port range>* is the range of ports you want to use.

For example:

```
ovconfchg -ns bbc.http -set CLIENT_PORT 14000-14003
```

Configure Reporter versions 3.6 and earlier, and OVPM 4.x as follows:

Edit the `/var/opt/OV/conf/BBC/default.txt` file as follows.

- 1 Locate the lines that apply to `CLIENT_PORT` and uncomment the line `;CLIENT_PORT =`.
- 2 Specify the port range for the `CLIENT_PORT` parameter. For example:

```
CLIENT_PORT = <port range>
```

Where *<port range>* is the range of ports you want to use. For example:

```
CLIENT_PORT = 14000-14003
```

Verify Firewall Configuration

To verify your configuration, use the command:

```
ovcodautl -ping -n <system name>
```

This output of this command indicate the status of your communication settings.

Configuring Systems with Multiple IP Addresses

If your environment includes systems with multiple network interfaces and IP addresses and you want to use a dedicated interface for the HTTP-based communication, then you can use the parameters `CLIENT_BIND_ADDR` and `SERVER_BIND_ADDR` to specify the IP address that should be used.

- If you have multiple network interfaces and IP addresses on the OVPA (Server) system, specify the `SERVER_BIND_ADDR` parameter as follows:

```
ovconfchg -ns bbc.http -set SERVER_BIND_ADDR <IP Address>
```
- If you have multiple network interfaces and IP addresses on the OVPM 5.0 (client) system, specify the `CLIENT_BIND_ADDR` parameter as follows:

```
ovconfchg -ns bbc.http -set CLIENT_BIND_ADDR <IP Address>
```
- If you have multiple network interfaces and IP addresses on the Reporter/OVPM 4.x system, specify the `CLIENT_BIND_ADDR` parameter.

Edit the `/var/opt/OV/conf/BBC/default.txt` file as follows:

- a Locate the lines that apply to `CLIENT_BIND_ADDR` and uncomment the line

```
;CLIENT_BIND_ADDR =
```
- b Specify the IP address for the `CLIENT_BIND_ADDR` parameter.

Configuring Secure Communication

OV Performance Agent supports certificate-based secure communication and client authentication based communication.

Using Certificates

OV Performance Agent supports certificate-based secure data communication only in the HP OpenView Operations 8.x environment.

To configure secure communication on your OVO setup, refer to the *HP OpenView Operations for UNIX Firewall Concepts and Configuration Guide*. For more information on OVO 8.x HTTPS agent, refer to the *HP OpenView Operations HTTPS Agent Concepts and Configuration Guide*.

If you have already configured HTTPS communication in the OVO 8.x environment, make the following changes to configure secure communication between OV Performance Agent and OVPM 5.0.



OV Reporter and OVPM 4.x do not support certificate-based secure communication.

On the OV Performance Agent system, set `SSL_SECURITY` to `REMOTE` for `coda`. Type the following commands:

```
ovconfchg -ns coda -set SSL_SECURITY REMOTE  
ovcodautl -config
```

Using Client Authentication

OV Performance Agent enables optional authentication of client connections from products such as OVPM or Reporter. The authentication capability allows you to specify, for a given OV Performance Agent instance, which hosts are allowed to make client connections to that instance.

The Client Authentication feature enables or disables connections from any version of the OVPM and Reporter clients. Your client software does not need to be updated for you to take advantage of this feature.

For authorized clients the authentication process is transparent — their client connection proceeds as it has with previous versions of OV Performance Agent. Unauthorized clients receive a message indicating denial of service.

Enabling Authentication with the `authip` File

Authentication is enabled by the presence of a file called `authip`. On systems where HTTP communication is enabled, the `authip` file exists in the `/var/opt/OV/conf/perf/` directory. The `authip` file lists hosts from which client connections are to be permitted.

- If the `authip` file exists in the `/var/opt/OV/conf/perf/` directory, its contents determine which hosts are permitted client connections. Clients running on the same host as the OV Performance Agent instance are automatically authenticated, which means the clients do not need an entry. A zero-length `authip` file dictates that only clients running on the OV Performance Agent host can connect.
- If the `authip` file does not exist (the default) in the `/var/opt/OV/conf/perf/` directory, then no authentication is performed and any client will be allowed to connect, as was the case with prior OV Performance Agent versions.

The `authip` file is checked each time a client attempts to connect for service with OV Performance Agent. OV Performance Agent does not need to be restarted for changes to the `authip` file to become effective.

The OV Performance Agent client authentication capability requires that your network be able to resolve the client entries in the `authip` file. Depending upon the nature of the entries, this may require name services such as those provided by DNS, NIS, or `/etc/hosts/` files.

A good test is to ensure that you can successfully ping each `authip` entry from the OV Performance Agent host. Client authentication works through a firewall with the same proviso that the client entries in the `authip` file be pingable from the OV Performance Agent host.

Formatting the `authip` File

The `authip` file must conform to the following format:

- One client host may be listed per line.
- Client entries can be in any one of the following formats:

- Fully qualified domain name
- Alias
- IP address (must be in IPv4 dotted quad format)
- Client entries can have no embedded spaces.
- A line containing a # in the first column is taken as a comment, and is ignored.
- Blank or zero-length lines are ignored.
- The IP address may not have a leading zero. For example, the IP address 23.10.10.10 cannot be represented as 023.10.10.10.

Thus, given an `/etc/hosts` entry as follows:

```
123.456.789.1 testbox testbox.group1.thecompany.com
```

any one of the following entries in the `authip` file would enable clients from the `testbox` host to connect:

```
#===== Examples of authip file entries =====
#
# Use of an IP address
123.456.789.1

#
# Use of an alias
testbox

#
# Use of a fully qualified domain name
testbox.group1.thecompany.com

#===== End of examples of authip file entries =====
```


Configuring Data Sources

OV Performance Agent uses the `coda` daemon to provide collected data to the alarm generator and analysis product. Each data source consists of a single log file set. The data source list that `coda` will access is maintained in the `datasources` configuration file that resides in the `/var/opt/OV/conf/perf/` directory. When you first start up OV Performance Agent after installation, a default data source named `SCOPE` is already configured and provides access to the active `scopeux` log file set.

If you want to add other data sources, you configure them in the `datasources` file. Then, when you restart OV Performance Agent, the `coda` daemon reads the `datasources` file and makes the data available over `datacomm` linkages to analysis tools for each data source it finds.

Datasources Configuration File Format

Each entry you place into the `datasources` configuration file represents a data source consisting of a single log file set. The entry specifies the data source name and location. Fields are case-insensitive except for the log file path name. The syntax is:

```
datasource=datasource_name logfile=logfile_set
```

- **datasource** is a keyword. **datasource_name** is the name used to identify the data source. For example, the data source name used in alarm definitions or by analysis software. Data source names must be unique. They are translated into upper case. The maximum length for a data source name is 64 characters.
- **logfile** is a keyword. **logfile_set** is the fully-qualified name identifying the log file (created by the `dsilog` or `scope` process, ending in `.log`), and is case-sensitive.

Following are two examples of the `datasources` file's data source entries:

```
datasource=SCOPE logfile=/var/opt/perf/datafiles/logglob
datasource=ASTEX logfile=/tmp/dsidemo/log/astex/ASTEX_SDL
```

After updating `datasources`, use the `ovpa restart server` command to restart the `coda` daemon. First, `ovpa restart server` stops all `coda` processes. It then starts the `coda` daemon that rereads the `datasources` file and picks up any changes made to it.

Note that stopping the `coda` daemon processes results in any current connection in OV Performance Manager being lost. For example, if you are drawing a graph on a data source and try to draw another graph, you must reselect the data source in OV Performance Manager and re-establish the connection once the `coda` daemon is started again.

Examine the contents of the `/var/opt/OV/log/coda.txt` file to check if the `coda` daemon was activated or for error messages.

Deleting Data Sources

If you no longer want to view the log file data from OV Performance Manager or process alarms for the log file, you can modify the `datasources` file in the OV Performance Agent configuration directory, `/var/opt/OV/conf/perf/` to remove the data source and path to the log file set. Then run the `ovpa` program to restart the `coda` daemon as described in [Configuring Data Sources](#) above.

You can also remove the log file set if you no longer need the data. If you remove the log file set but do not remove the data source from `datasources`, `coda` will skip the data source.

You might also choose to stop logging data to a log file set but keep the `coda` daemon open so you can view the historical data in OV Performance Manager. In this case, stop the `dsilog` process but don't delete the data source from the `datasources` file.

Parm File

The `parm` file is a text file that specifies configuration of the `scopeux` data collector including log file maximum sizes, interesting process threshold definitions, and application definitions. Comments in the file provide an overview of the various settings.

The `parm` file is provided with OV Performance Agent in the `/opt/perf/newconfig/` directory and is copied into the `/var/opt/perf/` directory during installation. For a complete description of the `parm` file and its parameters, see the “Parm File” section in Chapter 2 of the *HP OpenView Performance Agent for UNIX User's Manual*.

Defining Alarms

If you plan to use alarms to monitor performance, you must specify the conditions that generate alarms in a set of alarm definitions in the OV Performance Agent `alarmdef` file. When OV Performance Agent is first installed, the `alarmdef` file contains a set of default alarm definitions. You can use these default definitions or customize them to suit your needs.

For instructions on defining alarms, see the “Performance Alarms” chapter in the *HP OpenView Performance Agent for UNIX User’s Manual*. This chapter also describes the alarm definition syntax, how alarms work, and how alarms can be used to monitor performance.

3 Documentation

This chapter describes the online and printable documentation for OV Performance Agent in the following topics:

- [OV Performance Agent Documentation Set](#)
- [Viewing Documents on the Web](#)
- [Adobe Acrobat Files](#)

OV Performance Agent Documentation Set

OV Performance Agent software includes the standard OV Performance Agent documentation set in viewable and printable file formats. You can view the Adobe Acrobat format (*.pdf) documents online and print as needed. ASCII text (*.txt) documents are also printable. However, you can view a text file on your screen using any UNIX text editor such as vi. The documents are listed in the following table along with their file names and online locations.

Table 2 OV Performance Agent Documentation Set

Document	File Name	Location
<i>HP OpenView Performance Agent for Linux Systems Installation and Configuration Guide</i>	ovpainst.pdf	/opt/perf/ paperdocs/ovpa/C
<i>HP OpenView Performance Agent for UNIX User's Manual</i>	ovpausers.pdf	/opt/perf/ paperdocs/ovpa/C
<i>HP OpenView Performance Agent for UNIX Data Source Integration Guide</i>	ovpadsi.pdf	/opt/perf/ paperdocs/ovpa/C
<i>HP OpenView Performance Agent for UNIX and GlancePlus Tracking Your Transactions</i>	tyt.pdf	/opt/perf/ paperdocs/arm/C
<i>Application Response Measurement (ARM) API Guide</i>	armapi.pdf	/opt/perf/ paperdocs/arm/C
<i>HP OpenView Performance Agent for Linux Metrics Definitions</i>	metlinux.txt metlinux.htm	/opt/perf/ paperdocs/ovpa/C
<i>OVPA metrics list by Data Class for all operating systems</i>	mettable.txt	/opt/perf/ paperdocs/ovpa/C

Viewing Documents on the Web

The listed .pdf and .htm documents can also be viewed on the HP OpenView Manuals web site at:

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

Select **Performance Agent** from the product list box, select the release version, select the OS, and select the manual title. Click [**Open**] to view the document online, or click [**Download**] to place the file on your computer.

Adobe Acrobat Files

The Adobe Acrobat files were created with Acrobat 7.0 and are viewed with the Adobe Acrobat Reader versions 4.0 and later. If the Acrobat Reader is not in your Web browser, you can download it from Adobe's web site:

<http://www.adobe.com>

While viewing a document in the Acrobat Reader, you can print a single page, a group of pages, or the entire document.

From Linux, you can read a .PDF file by entering:

acroread filename.pdf

A Configuring Coda

Introduction

This appendix provides a list of options to configure `coda` and the BBC communication broker `ovbbccb`, using the `ovconfchg` tool to change the configuration settings for OV Performance Agent.

DISABLE_PROSPECTOR

Use this option to specify the data collection preferences through `coda`, when both the OVO agent and OVPA are installed. The default value is `false`. The format is as follows:

```
ovconfchg -namespace coda -set DISABLE_PROSPECTOR <value>
```

- `true`: `coda` will not collect data for the Coda datasource
- `false`: `coda` will collect the data for the Coda datasource

RESPONSE_SIZE_LIMIT

Use this command to specify the maximum amount of memory allocated by the `coda` daemon for a query response. The default value is 104857600 (100 megabytes). The format is as follows:

```
ovconfchg -namespace coda -set RESPONSE_SIZE_LIMIT <value>
```



If the specified limit is exceeded the following error message appears:

```
coda_out_of_resource
```

SSL_SECURITY

Use this option to enable secure communication through `coda`. The default value is `NONE`. The format is as follows:

```
ovconfchg -namespace coda -set SSL_SECURITY <value>
```

- NONE: coda does not require SSL connections for either the local or remote clients
- REMOTE: coda requires all remote connections to use SSL
- ALL: coda requires all connections (both local and remote) to use SSL

SERVER_BIND_ADDR

Use this option to specify the bind address for the server port. The default value is `localhost`. When the value is set to `localhost`, all clients connect to `ovbbcch` and requests are forwarded by `ovbbcch` to `coda`. The format is as follows:

```
ovconfchg -namespace coda.comm -set SERVER_BIND_ADDR localhost
```

LOG_SERVER_ACCESS

You can enable or disable the access to server using this option. If this option is set to `true`, the communication broker BBC records every access to the server, providing information about the senders IP address, requested HTTP address, requested HTTP method, and response status. This value typically will not be changed.

```
ovconfchg -namespace coda.comm -set LOG_SERVER_ACCESS false
```

PROXY

Use this option to configure the proxy. The format is as follows:

```
ovconfchg -ns bbc.http -set PROXY proxy:port+(a)-(b)
```

Where, the variables *a* and *b* are comma separated lists of hostnames, networks, and IP addresses that apply to the proxy.

Multiple proxies may be defined for one PROXY key using the “;” or “,” delimiter. “-” before the list indicates that those entities do not use this proxy, “+” before the list indicates that those entities use this proxy. The first matching proxy is used.

For example:

```
ovconfchg -ns bbc.http -set PROXY srv1.abc.com:8088+*  
ovconfchg -namespace coda.comm -set SERVER_PORT 0
```

SERVER_PORT

You can configure the port settings of the default port used by the the BBC communication broker. Use the `ovconfchg` tool to change the port settings on the OV Performance Agent system. Type the commands:

```
ovconfchg -ns bbc.cb.ports -set SERVER_PORT <port number>
```

```
ovconfchg -namespace coda.comm -set SERVER_PORT <port number>
```

Glossary

alarm

An indication of a period of time in which performance meets or exceeds user-specified alarm criteria. Alarm information can be sent to an analysis system (such as OVPM) and to OVO. Alarms can be identified in historical data log files using the utility program.

alarm generator

Handles the communication of alarm information. It consists of `perfalarm` and the `agdb` database. The `agdb` database contains a list of OV Performance Manager analysis nodes (if any) to which alarms are communicated, and various on/off flags that you set to define when and where the alarm information is sent.

alarmdef file

The file containing the alarm definitions in which alarm conditions are specified.

application

A user-defined group of related processes or program files. Applications are defined so that performance software can collect performance metrics for and report on the combined activities of the processes and programs.

application log file

See logappl.

coda daemon

A daemon that provides collected data to the alarm generator and analysis product data sources including `scopeux` log files or DSI log files. `coda` reads the data from the data sources listed in the `datasources` configuration file.

data source

Consists of one or more classes of data in a single `scopeux` or DSI log file set. For example, the OV Performance Agent SCOPE data source is a `scopeux` log file set consisting of global data. See also **datasources file**.

datasources file

A configuration file residing in the `/var/opt/OV/conf/perf/` directory. Each entry in the file represents a `scopeux` or DSI data source consisting of a single log file set. See also **perflbd.rc**, **coda** and **data source**.

datasources configuration file

A configuration file residing in the `/var/opt/OV/conf/perf/` directory. Each entry in the file represents a `scopeux` or DSI data source consisting of a single log file set.

data source integration (DSI)

The technology that enables OV Performance Agent to receive, log, and detect alarms on data from external sources such as applications, databases, networks, and other operating systems.

default.txt

A communications configuration file used to customize communication parameters for HP OpenView applications.

device

A device is an input and/or output device connected to a system. Common devices include disk drives, tape drives, CD-ROM drives, printers, and user terminals.

device log file

See **logdev**.

DSI

See **data source integration**.

DSI log files

Log files containing self-describing data that are created by OV Performance Agent's DSI programs.

extract

An OV Performance Agent program that allows you to extract (copy) data from raw or previously extracted log files and write it to extracted log files. It also lets you export data for use by analysis programs.

extracted log file

A log file created by the `extract` program. It contains user-selected data ranges and types of data. An extracted log file is formatted for optimal access by the workstation analysis tool, OVPM. This file format is suitable for input to the `extract` and utility programs and is the preferred method for archiving performance data.

Glance

Glance (or GlancePlus) is an online diagnostic tool that displays current performance data directly to a user terminal or workstation. It is designed to assist you in identifying and troubleshooting system performance problems as they occur.

global

A qualifier that implies the whole system.

global log file

See **logglob**.

interesting process

A process becomes interesting when it is first created, when it ends, and when it exceeds user-defined thresholds for CPU use, disk use, response time, and so on.

log file set

A collection of files that contain data collected from one source.

logappl

The raw log file that contains measurements of the processes in each user-defined application.

logdev

The raw log file that contains measurements of individual device (such as disk and `netif`) performance.

logglob

The raw log file that contains measurements of the system-wide, or global, workload.

logindx

The raw log file that contains additional information required for accessing data in the other log files.

logproc

The raw log file that contains measurements of selected “interesting” processes. A process becomes interesting when it is first created, when it ends, and when it exceeds user-defined thresholds for CPU use, disk use, response time, and so on.

logtran

The raw log file that contains measurements of transaction data.

midaemon

An OV Performance Agent program that translates trace data into Measurement Interface counter data using a memory based MI Performance Database to hold the counters. This database is accessed by collector programs such as `scopeux`.

ovbbccb

The OpenView Operations Communication Broker for HTTP(S) based communication controlled by `ovc`. See also `coda` and `ovc`.

ovc

The OpenView Operations controlling and monitoring process. In a standalone OVPA installation, `ovc` monitors and controls `coda` and `ovbbccb`. If OVPA is installed on a system with OpenView Operations for UNIX 8.x agent installed, `ovc` also monitors and controls OpenView Operations for UNIX 8.x processes. See also `coda` and `ovbbccb`.

ovpa script

The OV Performance Agent script that has options for starting, stopping and restarting OV Performance Agent processes such as the `scopeux` data collector, alarm generator, `ttd`, `midaemon`, and `coda`. See also the `ovpa man` page.

OV Performance Manager (OVPM)

Provides integrated performance management for multi-vendor distributed networks. It uses a single workstation to monitor environment performance on networks that range in size from tens to thousands of nodes.

parm file

An OV Performance Agent file containing the parameters used by `scopeux` to customize data collection.

perflbd.rc

A configuration file residing in the `/var/opt/perf/` directory. This file is maintained as a symbolic link to the `datasources` file. See also **datasources file**.

perfstat

A program that displays the status of all performance processes in your system.

process

Execution of a program file. It can represent an interactive user (processes running at normal, nice, or real-time priorities) or an operating system processes.

process log file

See **logproc**.

raw log file

Summarized measurements of system data collected by `scopeux`. See **logappl**, **logproc**, **logdev**, **logtran**, and **logindx**.

real time

The actual time in which an event takes place.

resize

Changes the overall size of a log file using the utility program's `resize` command.

run file

Created by the `scopeux` collector to indicate that the `scopeux` process is running. Removing the run file causes `scopeux` to terminate.

scopeux

The OV Performance Agent data collector program that collects performance data and writes (logs) it to raw log files for later analysis or archiving. *See also raw log files.*

scopeux log files

See raw log files.

status.scope file

Created by the `scopeux` collector to record status, data inconsistencies, and errors.

system ID

The string of characters that identifies your system. The default is the host name as returned by `uname -n`.

transaction log file

See logtran.

transaction tracking

The technology used in OV Performance Agent that allows information technology (IT) resource managers to measure end-to-end response time of business application transactions.

ttd.conf

The transaction configuration file where you define each transaction and the information to be tracked, such as transaction name, performance distribution range, and service level objective.

utility

An OV Performance Agent program that allows you to open, resize, scan, and generate reports on raw and extracted log files. You can also use it to check `parm` file and `alarmdef` file syntax, and obtain alarm information from historical log file data

Index

A

- alarm generator, starting, 20
- authip file, 31
 - examples, 32
 - formatting, 31

C

- client authentication, 31
- coda.log file, 34
- coda daemon, 33
- configuring, 33
 - data sources, 33

D

- data sources, 33
 - deleting, 34
 - DSI, 33
 - SCOPE, 33
 - SCOPE default data source, 33
 - scopeux, 33
- datasources configuration file, 33
 - format, 33
- deleting data sources, 34
- disk space, requirements, 9
- documentation, 39
 - Adobe Acrobat files, 42
 - OVPA documentation set, 40
 - viewing on the web, 40

- DSI data sources, 33

E

- environment variables, 21
- examples directory
 - README, 22
- extract program, 33

F

- files
 - Adobe Acrobat, 42
 - authip, 31
 - coda.log, 34
 - datasources, 33
 - OVPA startup configuration, 21
 - parm, 35
 - status.scope, 22
- firewall
 - communicating across, 23
 - overview of OVPA communications
 - configuration, 24
 - systems with multiple IP addresses, 29
- firewall, communicating across, 23

I

- installation procedures
 - installing OVPA using OVO, 14
- installation requirements
 - hardware, 9

- installing
 - OVPA, 12, 14
 - procedures, 11
 - requirements, 9

O

- OV Operations
 - using to install OVPA, 14
- OVPA
 - installing, 12, 14
 - removing, 17
 - scripts, 20
 - starting, 20
 - starting and stopping, 20
 - startup configuration file, 21
 - status files, 22
 - stopping, 11
 - system startup and shutdown script, 21
- ovpa.install script, 13
- OVPA_START_COMMAND variable, 21
- OVPA_START variable, 21
- ovpa restart script, 20
- ovpa scripts
 - ovpa start, 20
- ovpa stop script, 12
- OV Performance Agent
 - installing using OVO, 14

P

- parm file, 35
- perfstat command, 11
- printable documentation files, 40

R

- removing OVPA, 17

S

- SCOPE default data source, 33
- scopeux
 - data sources, 33
 - starting, 20
- scopeux log file set, 33
- scripts
 - ovpa.install, 13
 - ovpa.remove, 17
 - ovpa restart alarms, 20
 - ovpa stop, 20
 - OVPA system startup and shutdown, 21
- starting
 - OVPA, 20
 - scopeux, 20
- starting and stopping OVPA, 20
- status files, 22
 - coda.log, 22
 - status.mi, 22
 - status.perfalarm, 22
 - status.scope, 22
 - status.ttd, 22
- stopping OVPA, 11
- stopping processes prior to installation, 11

V

- variables, 21
 - environment, 21
 - OVPA_START, 21
 - OVPA_START_COMMAND, 21