

HP Performance Agent

for the UNIX, Windows and Linux operating systems

Software Version: 5.00

Shared Components Reference Guide

Document Release Date: August 2009
Software Release Date: August 2009



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 2008-2009 Hewlett-Packard Development Company, L.P.

Trademark Notices

UNIX® is a registered trademark of The Open Group.

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

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

Documentation Updates

This guide's title page contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates, or to verify that you are using the most recent edition of a document, go to:

<http://h20230.www2.hp.com/selfsolve/manuals>

This site requires that you register for an HP Passport and sign-in. To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

You can visit the HP Software Support Online web site at:

<http://www.hp.com/go/hpsoftwaresupport>

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

HP Software Support Online provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the HP Software Support web 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:

http://h20230.www2.hp.com/new_access_levels.jsp

To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

Contents

1 Shared Components	3
Reference Section	3
Modifying Configuration Parameters	6
Viewing Configurations Settings	8
Tracing	9
Configuring BBC	12
Troubleshooting Shared Component related issues	14
Debugging BBC-based Servers	14
Starting and Stopping the Communication Broker	16
Managing HP Operations Control Service	20
Troubleshooting issues related to component registration	23
Managing Unique Node Identifier	25
Troubleshooting Coda	27

1 Shared Components

Performance Agent packages the following shared components:

- L-Core
 - HP Software Cross Platform Component
 - HP Software Security Core
 - HP Software HTTP Communication
 - HP Software Process Control
- Coda
 - HP Software Performance Access
 - HP Software Performance Core

Reference Section

HP Performance Agent includes specific components for some tasks such as communication, tracing to record application performance for troubleshooting. Listed below are commands and options which allow you to manage these shared components.

Communication Broker Port Namespace

`bbc.cb.ports` is the parameter that defines the list of ports for all Communications Brokers in the network that may be contacted by applications on this host. The default port number for all BBC CBs is 383. You can use the following parameters:

```
string PORTS
```

This configuration parameter must be the same on all nodes. To change the port number of a BBC CB on a particular host, the hostname must be added to this parameter, for example, `name.hp.com:8000`. You can use an asterisk “*” as a wild card to denote an entire network, for example; `*.hp.com:8001`. Note too, that either a comma “,” or a semicolon “;” should be used to separate entries in a list of hostnames, for example;

```
name.hp.com:8000, *.hp.com:8001.
```

In these examples, all hostnames ending in “hp.com” will configure their BBC Communication Broker to use port 8001 except host “name” which will use port 8000. All other hosts use the default port 383. You can also use IP addresses and the asterisk wild card (*) to specify hosts. For example;

```
15.0.0.1:8002, 15.*.*.*:8003
```

HTTP Namespace

`bbc.http` is the HTTP Namespace for node-specific configuration. For application-specific settings, see the section `bbc.http.ext.*`. Note that application-specific settings in `bbc.http.ext.*` override node-specific settings in `bbc.http`. You can use the following parameters:

Parameter	Description
<code>int SERVER_PORT = 0</code>	By default this port is set to 0. If set to 0, the operating system assigns the first available port number. This is the port used by the application <code><app_Name></code> to listen for requests. Note that it only really makes sense to explicitly set this parameter in the <code>bbc.http.ext.<app_Name></code> namespace, as the parameter is application specific with any other value than the default value.
<code>string SERVER_BIND_ADDR = <address></code>	Bind address for the server port. Default is <code>localhost</code> .
<code>string CLIENT_PORT = 0</code>	Bind port for client requests. This may also be a range of ports, for example <code>10000-10020</code> . This is the bind port on the originating side of a request. Default is port 0. The operating system will assign the first available port. Note that Microsoft Windows systems do not immediately release ports for reuse. Therefore on Microsoft Windows systems, this parameter should be a large range.
<code>string CLIENT_BIND_ADDR = <address></code>	Bind address for the client port. Default is <code>INADDR_ANY</code> .

Parameter	Description
bool LOG_SERVER_ACCESS = false	If this parameter is set to true, every access to the server is logged providing information about the sender's IP address, requested HTTP address, requested HTTP method, and response status.
string PROXY	<p>Defines which proxy and port to use for a specified hostname.</p> <p>Format:</p> <pre>proxy:port +(a)-(b);proxy2:port2+(a)-(b); ...;</pre> <p>a: list of hostnames separated by a comma or a semicolon, for which this proxy shall be used.</p> <p>b: list of hostnames separated by a comma or a semicolon, for which the proxy shall <i>not</i> be used.</p> <p>The first matching proxy is chosen.</p> <p>It is also possible to use IP addresses instead of hostnames so 15.*.*.* or 15:*:*:*:*:*:* would be valid as well, but the correct number of dots or colons MUST be specified. IP version 6 support is not currently available but will be available in the future.</p>
string DOMAIN	<p>This defines the default DNS domain to use if no domain is specified for a target host. This domain name will be appended to hostnames not containing a DNS domain name, if a match for the hostname alone cannot be found. This will be done for PROXY lookups and lookups in the [cb.ports] table, for example if the hostname "merlin" is specified and the DOMAIN = "bbn.hp.com", then the [cb.ports] entries will first be searched for the match of "merlin". If there is no match found for the hostname "merlin", then a search will be made for "merlin.bbn.hp.com", "*.bbn.hp.com", "*.hp.com", "*.com" and "*", in that order.</p>

Modifying Configuration Parameters

The command `ovconfchg` allows you to configure settings files, update the configuration database, and trigger notification scripts.

`ovconfchg` manipulates the settings in either the system-wide configuration file or the configuration file for the specified OV Resource Group, `local_settings.ini`, updates the configuration database, `settings.dat`, and triggers notification scripts. If `ovconfchg` is called without options, or only with `-ovrg`, no settings are changed but an update is triggered anyway. This is to allow updating after default settings files have been added, removed, or updated.

When `ovconfchg` runs, all configuration settings are read and merged in memory. Default definitions are used to make corresponding checks, as well as to emit and log warnings in the event of a violation. During this process, file locks are used to prevent parallel updates. A new configuration database is then created containing the merged data.

Synopsis

```
ovconfchg -h | -help
```

```
ovconfchg -version
```

```
ovconfchg [-ovrg <OVRG>] [-edit | -job {-ns namespace {-set <attr> <value> | -clear <attr> | -clear -all} ... } ... ]
```

Parameters

`ovconfchg` recognizes the following options:

<code>-h -help</code>	Displays all the options for the <code>ovconfchg</code> command.
<code>-version</code>	Displays the version of the <code>ovconfchg</code> command.
<code>-edit</code>	Starts a text editor to edit the settings file, <code>local_settings.ini</code> . The text editor used is determined by the <code>\$EDITOR</code> environment variable. If <code>\$EDITOR</code> is not set, <code>vi</code> starts on UNIX and Notepad starts on Windows.
A temporary copy of the file is created for editing. After the changes are made, the file is validated for syntax errors. The syntax rule for validation is that the namespace and attribute names should contain only letters (a-z, A-Z), digits (0-9), period(.) and underscore(_) characters.	If the validation fails, the line number of the error is reported and the user will be prompted to correct the file. If Yes, the file will be reopened for making the necessary changes. If No, the original settings file remains unchanged. If the validation is successful, the changes are saved into the original settings file.

-h -help	Displays all the options for the <code>ovconfchg</code> command.
Do not configure binary values using this option. This can corrupt the file. It is also recommended to restrict the data entered using this option to US-ASCII (7-bit only) subset.	Do not open the settings file directly in a text editor and change it. This can corrupt the file.
<code>-job</code>	Create an update job file only and do not synchronize.
<code>-ns -namespace <namespace></code>	Sets a namespace for the <code>-set</code> and <code>-clear</code> options.
<code>-set <attr> <value></code>	Sets an attribute value in the namespace specified by the <code>-namespace</code> option. The local or OV resource settings file is updated accordingly.
<code>-clear <attr></code>	Clears the local setting for the attribute <code>attr</code> in the namespace specified by the <code>-namespace</code> option. The local settings file is updated accordingly.
<code>-clear -all</code>	Clears all local settings. The local settings file is updated accordingly.

Files

The `ovconfchg` command uses the following files to store local settings:

- `<DataDir>/conf/xpl/config/local_settings.ini`
- `<ShareDir>/<OVRG>/conf/xpl/config/local_settings.ini`

The `ovconfchg` command uses the following files to store database configuration settings:

- `<DataDir>/datafiles/xpl/config/settings.dat`
- `<ShareDir>/<OVRG>/datafiles/xpl/settings.dat`

Examples

The following examples show how to use the `ovconfchg` command:

- To assign the value 12 to the attribute `COUNT`, and assign the value "red blue white" to the attribute `COLORS` in the namespace, `tst.lib`:

```
ovconfchg -ns tst.lib -set COUNT 12 -set COLORS "red blue white"
```
- To clear the attribute `COUNT` in the namespace `tst.lib`:

```
ovconfchg -ns tst.lib -clear COUNT
```
- To remove all locally configured attributes from the namespace `tst.lib`:

```
ovconfchg -ns tst.lib -clear '*'
```

- For the OV resource group server, assign the value 50 to the attribute COUNT in the namespace tst.lib:

```
ovconfchg -ovrg server -ns tst.lib -set COUNT 50
```

Viewing Configurations Settings

The command `ovconfget` returns specified attributes from the configuration database. Installed HP Software components have associated configuration settings files that contain one or more namespaces and apply system wide or for a specified OV Resource Group. A namespace is a group of configuration settings that belong to a component. All configurations specified in the settings files are duplicated in the `settings.dat` configuration database.

For each specified namespace, `ovconfget` returns the specified attribute or attributes and writes them to stdout. Used without arguments, `ovconfget` writes all attributes in all namespaces to stdout.

Synopsis

```
ovconfget -h | -help
ovconfget -version
ovconfget [-ovrg <OVRG>] [<namespace> [<attr>]]
```

Parameters

`ovconfget` recognizes the following options:

Parameters	Description
-h -help	Displays the options for the <code>ovconfget</code> command
-version	Displays the component version
<namespace> <attr>	Obtains the specified attribute in the specified namespace for the named OV Resource Group <OVRG> and writes them to stdout. If <i>namespace</i> is used without specifying an attribute, <attr>, <code>ovconfget</code> writes the contents of the database for the specified namespace. If neither <attr> nor <namespace> is specified, <code>ovconfget</code> writes the complete contents of the configuration database to stdout.

Files

The `ovconfget` command uses the following files to read configuration-database settings:

- <DataDir>/datafiles/xpl/config/settings.dat
- <ShareDir>/<OVRG>/datafiles/xpl/settings.dat

Examples

The following examples show how to use the `ovconfget` command:

- To return the value of the `Port` attribute in the `tst.settings` namespace, for example:
9012

```
ovconfget tst.settings Port
```

```
9012
```

- To return all attributes in the `tst.settings` namespace as multiple lines in the form of `attr=value`, for example:

```
ovconfget tst.settings
```

```
Port=9012
```

```
Protocols=HTTP FTP HTTPS
```

```
MaxFileSize=128
```

- To return all attributes in all namespaces on multiple lines, for example:

```
ovconfget
```

```
[tst.lib]
```

```
LibraryPath=/opt/OV/lib:/opt/OV/lbin/tst/var/opt/OV/tmp
```

```
[tst.settings]
```

```
Port=9012
```

```
Protocols=HTTP FTP HTTPS
```

```
MaxFileSize=128
```

Tracing

Tracing is used by the application to record information during program execution that may be helpful in analyzing process flow and behavior or debugging.

Tracing Tools

<code>ovtrccfg</code>	<code><OV Install Dir>/support/</code>
<code>ovtrcd</code>	<code><OV Install Dir>/lbin/xpl/trc/</code>
<code>ovtrcgui</code>	<code><OV Install Dir>/support/</code>
<code>ovtrcadm</code>	<code><OV Install Dir>/support/</code>
<code>ovtrcmon</code>	<code><OV Install Dir>/support/</code>

Enabling Tracing

You can enable tracing using a trace configuration file for specific applications and trace areas. Trace configuration files are ASCII text files that can be viewed or modified using a standard text editor. The trace GUI (available only on Windows) can also be used to save a trace configuration file.

Trace Configuration File Syntax

```
Syntax Version TCF Version 3.2
Application APP: "Application-name"
Troubleshooting coda 21
Sink SINK: File "Output-name"
"force=[1];maxfiles=[1..100];maxsize=[0..1000];"
SINK: Socket "node name" "node=<node name>";"
Trace TRACE: "Component-name" "Category-name" <keyword list>
```

Sample trace configuration file for Coda

```
TCF Version 3.2
APP: "coda"
SINK: File "c:\\temp\\coda.trc" "force=1;maxfiles=10;maxsize=10;"
TRACE: "coda.logger" "Proc" Info Warn Error Developer Verbose Location
TRACE: "coda.logger" "Trace" Info Warn Error Developer Verbose Location
TRACE: "coda.util" "Trace" Info Warn Error Developer Verbose Location
TRACE: "coda" "Proc" Info Warn Error Developer Verbose Location
TRACE: "coda" "Trace" Info Warn Error Developer Verbose Location
TRACE: "xpl.net" "Trace" Info Warn Error Developer Verbose Location
TRACE: "coda.localmesa" "Proc" Info Warn Error Developer Verbose Location
TRACE: "coda.localmesa" "Trace" Info Warn Error Developer Verbose Location
TRACE: "coda.scopeaccess" "Proc" Info Warn Error Developer Verbose
Location
TRACE: "coda.scopeaccess" "Trace" Info Warn Error Developer Verbose
Location
TRACE: "coda.mesa" "Proc" Info Warn Error Developer Verbose Location
TRACE: "coda.mesa" "Trace" Info Warn Error Developer Verbose Location
TRACE: "coda.mesarea" "Proc" Info Warn Error Developer Verbose Location
TRACE: "coda.mesarea" "Operation" Info Warn Error Developer Verbose
Location
TRACE: "coda.prospector" "Proc" Info Warn Error Developer Verbose Location
TRACE: "coda.mesainstances" "Trace" Info Warn Error Developer Verbose
Location
TRACE: "coda.dataaccess" "Proc" Info Warn Error Developer Verbose Location
```

Steps for Enabling Tracing

Listed below are steps to enable tracing on UNIX:

- 1 Execute the following command to verify that the trace server process is running:

```
ps -ef | grep ovtrcd
```

- 2 If the trace server is not running, start the trace server manually using the command

```
<InstallDir>/lbin/xpl/trc/ovtrcd
```

If trace server is not running currently, before starting, you must restart the application which you want to trace.

- 3 Create trace configuration file with required sink type as shown in above example.
- 4 Run the command:

```
ovtrcdm -a localhost e.g: ovtrcdm -a
```



Disable the clients using the command `ovtrcadm -d <list of clients>` after you finish tracing from those clients.

5 Run the command

```
ovtrccfg -server <server_name> <trace_config_file_name>
```

Where `server_name` is the system name where the trace server is running and `trace_config_file_name` refers to the trace configuration file as mentioned above.

6 If the SINK in your trace configuration file is Socket you can use `ovtrcmon` tool to monitor the tracing status using the command `ovtrcmon -server localhost`.

7 If the SINK in your trace configuration file is **FILE** then the trace files are generated in the specified location (on the system where the application is running) as mentioned in trace configuration file.

8 You can stop or disable tracing by running the following command:

```
ovtrccfg -server localhost off
```



It is recommended that you disable tracing if you do not want to use tracing and stop the trace server. This is because you must restart your application (after starting the trace server) when you need to start trace again. You must always run your trace server and you can disable it when it is not required.

Listed below are steps to enable tracing on Windows:

- 1 Open the Services window (or run “net start”) and verify that the state of the service named HP OpenView Shared Trace Service has started.
- 2 Start the Trace Wizard and select the option to load a configuration file using the following steps.
- 3 Start the `ovtrcgui` tool
- 4 From File menu, click **Trace Wizard** and then select **Next**.
- 5 Select the **Configure local applications by loading a saved configuration** option.
- 6 From the **Open** dialog, locate and select the trace configuration file.
- 7 By default, a Trace Monitor window is displayed after the configuration is completed. To open a new Trace Monitor window, use the following steps.
 - a From File menu, click **New** and then select **Trace Monitor**.
 - b Select the system you want to monitor trace messages and then click **OK**. This starts a new trace monitor window.
- 8 Stop tracing using the following steps:
 - a Select the configuration window associated with the tracing.
 - b From the **File** menu, click **Close**.



The trace server can be stopped using the command `ovtrcadm -srvshutdown` in UNIX and through service control manager in Windows. However, it is recommended that you must not forcefully kill the trace server as this might cause problems while starting the trace server again.

Configuring BBC

The configuration file of a node using HTTPS communication is `bbc.ini`. This file is available at: `<Installdir>/misc/xpl/config/defaults/`.

The `bbc.ini` file consists of sections headed by namespaces which contain the settings for each namespace. The `bbc.ini` file contains the namespaces listed below:

Communication Broker Namespace

`bbc.cb` is the communication broker namespace. You can use the following parameters to configure the communication broker namespace:

Parameter	Description
<code>string CHROOT_PATH = <path></code>	On UNIX systems only, the <code>chroot</code> path is used by the <code>ovbbccb</code> process. If this parameter is set, the <code>ovbbccb</code> process uses this path as the effective root thus restricting access to a limited part of the file system. This parameter is ignored on Microsoft Windows and Sun Solaris 7 systems. See the <code>chroot</code> man page for details on <code>chroot</code> . The default value is <code><OvDataDir></code> .
<code>bool SSL_REQUIRED = false</code>	If this parameter is set to <code>true</code> , the communication broker requires SSL authentication for all administration connections to the communication broker. If this parameter is set to <code>false</code> , non-SSL connections are allowed to the communication broker.
<code>bool LOCAL_CONTROL_ONLY = false</code>	If this parameter is set to <code>true</code> , the communication broker only allows local connections to execute administrative commands such as <code>start</code> and <code>stop</code> .

Parameter	Description
<code>bool LOG_SERVER_ACCESS = false</code>	If this parameter is set to <code>true</code> , every access to the server is logged providing information about the sender's IP address, requested HTTP address, requested HTTP method, and response status.
<code>int SERVER_PORT = 383</code>	By default this port is set to 383. This is the port used by the communication broker to listen for requests. If a port is set in the namespace <code>[bbc.cb.ports]</code> , it takes precedence over this parameter.
<code>string SERVER_BIND_ADDR = <address></code>	Bind address for the server port. Default is <code>INADDR_ANY</code> .

Troubleshooting Shared Component related issues

This section lists the commands and options which you can use to troubleshoot issues related to shared components while using Performance Agent.

Debugging BBC-based Servers

The `bbcutil` command helps you to debug a BBC-based server. The `bbcutil` command can be used to list all applications registered to a Communication Broker, to check whether specified communication services are alive, and to display details about the current state of the server.

Synopsis

```
bbcutil -h|-help
```

```
bbcutil -version
```

```
bbcutil -ping {[<hostname>|<ip>[:<port>]] | [<uri>]} [count]  
[-v|-verbose]
```

```
bbcutil -status {[<hostname>|<ip>[:<port>]] | [<uri>]} [-v|-verbose]
```

```
bbcutil -getcbport [<hostname>|<ip>]
```

Parameters

The `bbcutil` command incorporates the options in the following list. The syntax for the `[<hostname>|<ip>[:<port>]]` string.

Example; in the options `-registrations` or `-ping`, can be a hostname and a port separated by a colon (:) but can also be a full URL path (including protocol), such as:

```
https://merlin.guilford.mycom.com:383/com.hp.ov.coda
```

`bbcutil` recognizes the following options:

Parameters	Descriptions
-h -help	Displays and describes the available options for the <code>bbcutil</code> command.
-version	Displays the version of the HP Software communication in use.
-ping {[<hostname> <ip>][:<port>]] [<uri>]} [count]	Pings the specified HP Software server process. A hostname or IP address with an optional port number or a URL may be given to locate the server process to ping. If a URL is given with the path of a valid process registered with the Communications Broker, the Communications Broker will automatically forward the ping to the registered process. Count specifies the number of times to execute the ping. The node may be specified with a hostname or IP address. Default for the node is "localhost". Default for the port is the Communications Broker port on the specified node. Default count is 1.
-status {[<hostname> <ip>][:<port>]] [<uri>]}	Displays the status of the specified HP Software server process. A hostname or IP address with an optional port number or a URI may be given to locate the server process. The node may be specified with a hostname or IP address. Default for the node is localhost. Default for the port is the Communications Broker on the specified node.
-verbose	Shows more detailed output.
-getcbport [<hostname> <ip>]	Displays the configured Communications Broker port number of the node specified by <hostname> or <ip>. If the hostname or IP address is not specified, localhost is assumed. If no Communication Broker port number is configured for the node, the default value 383 is displayed.

Exit Status

The following exit values are returned:

0	bbcutil exited normally with no error.
1	Command syntax error encountered. See command syntax for more details on possible values.
2	Command partially succeeded.
3	Command failed. See command output for more detailed information.
4	bbcutil could not complete the requested command due to an authorization error.
100	An exception was encountered causing the Communications Broker to exit.

Corresponding error messages are written to stderr.

Examples

The following examples show you how to use the `bbcutil` command:

- To show the status of Communication Broker on the local node:

```
bbcutil -status
```
- To query the communication server located at `https://merlin.guilford.mycom.com:383/com.hp.ov.coda` for details about the current state of the server:

```
bbcutil -ping https://merlin.guilford.mycom.com:383/com.hp.ov.coda
```
- To get the IP address and Communications Broker port number of a target node `node1`

```
bbcutil -gettarget node1
```

Starting and Stopping the Communication Broker

The `ovbbccb` command allows you to control HTTPS communication using Communication Broker proxies on local nodes. It controls starting of the Communication Broker as a background daemon process or in normal mode, stopping, and re-initializing of the Communication Broker. `ovbbccb` is also used to start and stop OV resource groups in the Communication Broker.

`ovbbccb` can also be used to list all active OV resource groups and all applications registered to a Communication Broker, to check whether specified communication services are alive and to display details about the current state of the server.

Synopsis

```
ovbbccb -h|-help
```

```

ovbbccb -version
ovbbccb -daemon|-nodaemon [-debug] [-v|-verbose]
ovbbccb -kill|-reinit [<hostname>|<ip>] [-v|-verbose]
ovbbccb -ping {[<hostname>|<ip>[:<port>]] | [<uri>] [-v|-verbose]}
ovbbccb -status {[<hostname>|<ip>[:<port>]] | [<uri>] [-v|-verbose]}

```

Parameters

The `ovbbccb` command incorporates the options in the following list. The syntax for the `[<hostname>|<ip>][:<port>]` string, for example; in the options `-registrations` or `-ping`, can be a hostname and a port separated by a colon (:) but can also be a full URL path including protocol. for example:

`https://merlin.guilford.mycom.com:383/com.hp.ov.coda`

`ovbbccb` recognizes the following options:

Parameters	Description
<code>-h -help</code>	Displays and describes the available options for the <code>ovbbccb</code> command.
<code>-version</code>	Displays the version of the OV communication in use.
<code>-daemon</code>	Starts the Communication Broker either as a background daemon process on a UNIX machine or a service on a Microsoft Windows machine.
<code>-nodaemon</code>	Starts the Communication Broker as a foreground process (<i>default</i>).
<code>-debug</code>	Disable Control-C signal handler for debugging.
<code>-verbose</code>	Shows more detailed output.
<code>-start <ovrg></code> <code>[<hostname> <ip>]</code>	Starts the OV resource group specified by <code><ovrg></code> in the Communication Broker on the host specified by <code><hostname></code> or <code><ip></code> . If the hostname or IP is not specified, <code>ovbbccb</code> uses the local host as the host. You must configure the resource group on a cluster node to use this option.
<code>-kill [<hostname> <ip>]</code>	Stops the Communication Broker on the host specified by <code><hostname></code> or <code><ip></code> . If the hostname or IP is not specified, <code>ovbbccb</code> used the local host as the host. You must set the <code>LOCAL_CONTROL_ONLY</code> parameter to false to make this option work on a remote node.

Parameters	Description
-reinit [<hostname> <ip>]	<p>The Communication Broker specified in <hostname> or <ip> reloads the configuration data and is re-initialized. If the hostname or IP is not specified, ovbbccb uses the local host as the host.</p> <p>The SIGHUP signal may also be used on UNIX systems to re-initialize the Communication Broker process.</p> <p>You must set the LOCAL_CONTROL_ONLY parameter to false to make this option work on a remote node.</p>
-ping {[<hostname> <ip>[:<port>]] [<uri>]}	<p>Pings the specified HP Software server process. A hostname or IP address with an optional port number or a URI may be given to locate the server process to ping. If a URI is given with the path of a valid process registered with the Communication Broker, the Communication Broker will automatically forward the ping to the registered process. The node may be specified with a hostname or IP address. Default for the node is "localhost". Default for the port is the HP Software Communication Broker port on the specified node.</p>
-status {[<hostname> <ip>[:<port>]] [<uri>]}	<p>Displays the status of the specified HP Software server process. A hostname or IP address with an optional port number may be given to locate the server process. Default for the node is "localhost". Default for the port is the HP Software Communication Broker port on the specified node.</p>

Exit Status

The following exit values are returned:

0	ovbbccb exited normally with no error.
1	Command syntax error encountered. See command syntax for more details on possible values.
2	Command partially succeeded.
3	Command failed. See command output for more detailed information.
4	The Communications Broker start command failed because a Communications Broker process is already running

5 The Communications Broker failed to start because a Local Location Broker process is already running. The HP Software Communications Broker is not supported on systems running the LLB. Stop the LLB before attempting to start the Communications Broker.

6 The Communications Broker failed to stop because the Communications Broker process is already stopped.

7 The Communications Broker failed to start due to a bind exception on the Communications Broker port to be opened.

8 The Communications Broker could not complete the command due to an authorization error.

100 An exception was encountered causing the Communications Broker to exit.

Corresponding error messages are written to `stderr`.

Examples

The following examples show you how to use the `ovbbccb` command:

- To start the Communication Broker as a daemon process on the local system:

```
ovbbccb -daemon
```

- To display the status of the specified HP Software server process:

```
ovbbccb -status
```

The following output appears:

```
Status: OK
(namespace, Port, Bind Address, Open Sockets)
<default> 383 ANY 2HP OpenView HTTP Communication Incoming
Connections
To machine1.example.hp.com:
localhost:17282 76bb6662-2cd3-7531-1221-b67340fb721f BBC 06.10.209;
ovbbccb 06.10.209
```

```
HP OpenView HTTP Communication Reverse Channel Connections
Opened from machine1.example.hp.com:
machine:9090 BBC 06.10.209; ovbbccrcp 06.10.209 30 Jan 2009 14:53:29
265 ms
```

```
Failed from machine1.example.hp.com:
machine11:9090 Host Unknown 30 Jan 2009 14:53:28 5687 ms
machine13:9090 Host Unknown 30 Jan 2009 14:53:28 5687 ms
```

```
Pending from machine1.example.hp.com:
machine14: Trying to connect 30 Jan 2009 14:53:28 (11312) ms
machine12:9090 Trying to connect 30 Jan 2009 14:53:28 (5671) ms
```

Managing HP Operations Control Service

The `ovc` command allows you to perform actions on local components. `ovc` controls the starting and stopping, event notification, and status reporting of all components registered with the HP Operations Control service.

A component can be a server process belonging to any of the products such as HP Operations Manager for Windows, HP Operations agents (for example, the Performance Agent or the Discovery Agent), an event interceptor, or an application delivered by an integrator.

Synopsis

```
ovc -h|-help
ovc -start [<target> ... ] [-boot]{[-async]|[-verbose]}
ovc -stop [<target> ... ][-nostart]{[-async]| [-verbose]}
ovc -restart [<target> ... ]
ovc -kill [-verbose]
ovc -status [<target> ... ] [-level <level>]
ovc -version
```


ovc recognizes the following options:

Parameters	Description
-h -help	Displays <i>all</i> available options for the ovc command.
-start [<target> ...] [-boot]{[-async] [verbose]}	Starts the selected components. <target> specifies a component or category. If <target> is not used, all components are started. If -boot is used, only components that start at boot time are started. The -async option starts the components asynchronously. If you use the -verbose option, ovc command displays the progress of the command execution. You can use the -async or the -verbose option, but you must not include these options together in a command.
-stop [<target> ...] [-nostart]{[-async] [ver bose]}	Stops the selected components. <target> specifies a component or category. If <target> is not used, all components are stopped <i>except</i> components, which belong to the CORE component group. If you specify the -nostart option and if the control daemon is not running, the command does not perform any action. If you do not specify the -nostart option, the ovc -stop command starts the control daemon and ovbbccb components if these components are not running. The -async option starts the components asynchronously. If you use the -verbose option, the ovc command displays the progress of the command execution. You can use the -async or the -verbose option, but you must not include these options together in a command.
-restart [<target> ...]	Stops components before they are restarted. <target> specifies a component or category. If <target> is not used, all components are stopped and restarted.

Parameters	Description
-kill [-verbose]	Stops all components registered with the HP Operations Control service. If you use the <code>-verbose</code> option, the <code>ovc</code> command displays the progress of the command execution.
-status [<target> ...]	Reports the status of a component or category specified by <target>. The status report contains the component's label, description, category, process ID, and STATE. Components can be in state: Stopped (0 in numeric format), Starting (1), Initializing (2), Running (3), Stopping (4), N/A (5) or Aborted (6). If <target> is not specified, the status of <i>all</i> components is returned.
-version	Prints the version of <code>ovc</code>

Exit Status

The following exit values are returned:

0	Success.
1	Not defined.
2	Ignored.
62	The UNIX daemon or Windows service is not running.
63	The Control daemon is being initialized.
64	Generic error.
65	Invalid target.
67	Operation aborted.
69	Missing prerequisite.
70	Authorization error.
71	Operation on prerequisite failed.
73	Invalid event.

Examples

The following examples show how to use the `ovc` command and some of its options to control and display important information about registered components.

- To start the component registered as `coda`:

```
ovc -start coda
```

Before `coda` itself starts, all the components that `coda` depends on are started.

- To start the component registered as `coda` and display the progress of the command execution:

```
ovc -start coda -verbose
```

Before `coda` itself starts, all the components that `coda` depends on are started.

- To print the status of all registered components:

```
ovc -status
```

- To stop the component registered as `coda`:

```
ovc -stop coda -verbose
```

Before `coda` itself stops, all the components that depend on `coda` are stopped. This command starts the control daemon and `ovbbccb` components if these components are not running.

- To stop the component registered as `coda` using the `ovc -stop[<target>...] -nostart` option:

```
ovc -stop coda -nostart
```

Before `coda` itself stops, all the components that depend on `coda` are stopped. This command does not perform any action if the control daemon is not running.

- To start all components (and their dependents) belonging to category `SERVER` and `AGENT`.

```
ovc -start SERVER AGENT
```

- To print the status of the component `coda` and display the registration details:

```
ovc -status coda -level 2
```

Troubleshooting issues related to component registration

`ovcreg` is used to register a component with (and de-register the component from) the `OvCtrl`. The `ovcreg` command can also be used to check a component registration file for syntactical correctness.

If the `OvCtrl` daemon (`ovcd`) is running at the time of registration, it will be informed about the new component only if the `-add` option was applied and the component is not started. The `OvCtrl` shows the new component the next time the `ovc` command is called with the `-status` option.

If the `OvCtrl` daemon (`ovcd`) is running, the component will be stopped if the `-del(ete)` option was applied. NOTE: this option will *not* stop CORE components, which are denoted by the option `CoreProcess` in the registration file. CORE components should be stopped with `ovc` command and the `-kill` option.

Synopsis

```
ovcreg -h|-help
```

```
ovcreg -check [<filename>]
```

```
ovcreg -add [<filename>]
```

```
ovcreg -del [<component>]
```

```
ovcreg -version
```

Parameters

ovcreg recognizes the following options:

-h -help	Displays <i>all</i> available options for the ovcreg command.
-check [<i><filename></i>]	Checks the syntax of <i><filename></i> . <i><filename></i> must not contain more than one component.
-add [<i><filename></i>]	Checks the syntax of <i><filename></i> and stores a copy in the configuration directory. Adding a component with a name which is already registered with the OvCtrl will overwrite the original registration with the new one. <i><filename></i> must not contain more than one component.
-del [<i><component></i>]	Stops and de-registers the specified <i><component></i> from the OvCtrl and deletes the specified <i><component></i> registration file. NOTE: the delete option does not stop CORE components.
-version	Displays the version of ovcreg

Exit Status

The following exit values are returned:

0	Success - The syntax of the file is correct and the registration file is successfully added or deleted.
1	Wrong usage
2	Parsing error
3	Error deleting registration file
5	Error writing XML file
6	Component is not registered
7	Error stopping component
8	Error deleting component

Files

Registration files for components registered with the OvCtrl for the supported platforms reside in the following locations:

- AIX, HP-UX, Linux, Solaris:
`/var/opt/OV/conf/ctrl/*.xml`
- True64:
`/usr/var/opt/OV/conf/ctrl/*.xml`
- Microsoft Windows:
`C:\Program Files\HP\HP BTO Software\conf\ctrl*.xml`

Note that the user can change the specified default location for the registration files on machines running Microsoft Windows.

Examples

The following examples show how to use the `ovcreg` command and some of its options to control and display important information about registered components.

- To check the syntax of the component registration file: `opcle.xml`:
`ovcreg -check opcle.xml`
- To check the syntax of the component registration file, `opcle.xml`, and add the component defined in the component registration file, `opcle.xml` to the OvCtrl:
`ovcreg -add opcle.xml`
- To stop and de-register the component registered as `opcle`:
`ovcreg -del opcle`

Managing Unique Node Identifier

Use this command if the Core ID is not set by default. The command `ovcoreid` manages the unique node identifier `OvCoreId` on the local node. The `ovcoreid` command is used to display existing `OvCoreId` values and, in addition, create and set new `OvCoreId` values on the local node.

Synopsis

```
ovcoreid -show [-ovrg <OV_Resource_Group>]
ovcoreid -create [-force] [-ovrg <OV_Resource_Group>]
ovcoreid -set <OvCoreId> [-force] [-ovrg <OV_Resource_Group>]
ovcoreid -version
ovcoreid -h|-help
```

Parameters

The `ovcoreid` command accepts the following parameters and options:

<code>-show [-ovrg <OV_Resource_Group>]</code>	Displays the current <code>OvCoreId</code> of the system (configuration setting <code>CORE_ID</code> in namespace <code>[sec.core]</code>). This is the default if no parameters are specified. If the <code>OvCoreId</code> you want to show belongs to an OpenView Resource Group, use the <code>-ovrg</code> option to specify the name of the Resource Group. If an OV Resource Group is specified, the corresponding configuration settings will be read or modified as well. If you specify a non-existent OV Resource Group, <code>ovcoreid</code> displays the local <code>OvCoreId</code> .
<code>-create [-force] [-ovrg <OV_Resource_Group>]</code>	Generates a new <code>OvCoreId</code> . If a <code>CORE_ID</code> value already exists, the existing <code>OvCoreId</code> is only overridden when <code>-force</code> is specified. If the <code>OvCoreId</code> you want to show belongs to an OpenView Resource Group, use the <code>-ovrg</code> option to specify the name of the Resource Group. If an OV Resource Group is specified, the corresponding configuration settings will be read or modified as well. If you specify a non-existent OV Resource Group, <code>ovcoreid</code> displays an error.
<code>-set [-force] [-ovrg <OV_Resource_Group>]</code>	Sets a specific <code>OvCoreId</code> . The <code>-force</code> option must be used if an <code>OvCoreId</code> value has already been set. If the <code>OvCoreId</code> you want to show belongs to an OpenView resource group, use the <code>-ovrg</code> option to specify the name of the resource group. If an OV Resource Group is specified, the corresponding configuration settings will be read or modified as well.
<code>-version</code>	Returns the version of the tool (the component version).
<code>-h -help</code>	Display all available command options.

Exit Status

The following exit values are returned:

0	All steps were successful.
1	If <code>-create</code> or <code>-set</code> is used without <code>-force</code> and a value for <code>OvCoreId</code> already exists.
2	One or more steps were not successful.

Corresponding error messages are written to `stderr`.



Changing the `OvCoreId` of a system is analogous to giving the system a new identity and is an action that should only be executed if the consequences are fully understood. Changing the `OvCoreId` of a system requires a number of significant changes including the need for a new certificate, and having to do appropriate reconfiguration of the HP Software server(s).

Examples

The following examples show you how to use the `ovcoreid` command:

- To display the `OvCoreId` for the local node:
`ovcoreid -show`
- To create and set a new `OvCoreId` on the local node:
`ovcoreid -create`
- To set the specified `OvCoreId` on the local node:
`ovcoreid -set <OvCoreId>`

Troubleshooting Coda

If coda is not responding or if the coda process is aborted you can verify the status files. The coda status files provide information about errors and warnings that may be encountered in coda processes. They are formatted in plain ASCII text.

For UNIX/Linux:

```
/var/opt/OV/log/coda.txt  
/var/opt/OV/log/System.txt
```

For Windows:

```
<Install Directory>\HP OpenView\data\log\coda.txt  
<Install Directory>\HP OpenView\data\log\System.txt
```

General Coda Troubleshooting

Check the hardware and software requirements. Check the release notes of the product consuming Coda for known problems & workarounds. Check `coda.log` file for error messages:

```
<OvAgentDataDir>/log/coda.log
```

- 1 Run “`codautl -status`” and verify Coda is running.

2 Run “codautl –support” to verify prospector is logging

If Coda is running and logging data, but it is not reporting any data, start tracing. For more information, see [Tracing](#) on page 9.

The `ovcodal util` command

This commands allows you to troubleshoot coda:

syntax: `./ovcodautl [options]`

Options:

- status: Get collector status.
- support: Test collector.
- config: Reconfigure.
- update: Reconfigure coda if the configuration database has changed.
- l10n: Localize output.
- obj: Display object model.
- dumpds <value>: Dump last record.
- ping: Ping the OvBbcCb and Coda.
- https: Use secure communications (https)
- verbose: Verbose output
- n <value>: Sets node name.
- migrate <value>: Migrate coda or client configuration.

We appreciate your feedback!

If an email client is configured on this system, by default an email window opens when you click on the bookmark “Comments”.

In case you do not have the email client configured, copy the information below to a web mail client, and send this email to **docfeedback@hp.com**

Product name:

Document title:

Version number:

Feedback:

