

# HP Network Node Manager i Software

For the Windows<sup>®</sup>, HP-UX, Linux, and Solaris operating systems

Software Version: 9.0x Patch 3

---

## Reference Pages

Document Release Date: March 2011  
Software Release Date: March 2011



## 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 2008–2011 Hewlett-Packard Development Company, L.P.

### Trademark Notices

Acrobat® is a trademark of Adobe Systems Incorporated.

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

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

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

### Oracle Technology — Notice of Restricted Rights

Programs delivered subject to the DOD FAR Supplement are 'commercial computer software' and use, duplication, and disclosure of the programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, programs delivered subject to the Federal Acquisition Regulations are 'restricted computer software' and use, duplication, and disclosure of the programs, including documentation, shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software-Restricted Rights (June 1987). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

For the full Oracle license text, see the license-agreements directory on the NNMi product DVD.

### Acknowledgements

This product includes software developed by the Apache Software Foundation.  
(<http://www.apache.org>)

This product includes software developed by the Indiana University Extreme! Lab.  
(<http://www.extreme.indiana.edu>)

This product includes software developed by The Legion Of The Bouncy Castle.  
(<http://www.bouncycastle.org>)

This product contains software developed by Trantor Standard Systems Inc..  
(<http://www.trantor.ca>)

# Network Node Manager Reference Pages

User Commands (1)	Administrator Commands (1M)	File Formats (4)
<a href="#">nmecatgets</a> <a href="#">nmmcluster</a> <a href="#">nmm.envvars</a> <a href="#">nmmprops</a> <a href="#">nmmsetcmduserpw.ovpl</a> <a href="#">nmmtrapdump.ovpl</a> <a href="#">nmmfindattachedswport.ovpl</a> <a href="#">nmmversion.ovpl</a> <a href="#">ovjbosspath.ovpl</a> <a href="#">ovjrepath.ovpl</a> <a href="#">ovstatus</a> <a href="#">nmmdumpevents</a> <a href="#">nmsnmpnotify.ovpl</a>	<a href="#">nmsdbmgr</a> <a href="#">nmmaction</a> <a href="#">nmmbackup.ovpl</a> <a href="#">nmmbackupembdb.ovpl</a> <a href="#">nmmcertmerge.ovpl</a> <a href="#">nmmchangedbpw.ovpl</a> <a href="#">nmmchangeembdbpw.ovpl</a> <a href="#">nmmchangesyspw.ovpl</a> <a href="#">nmmcommconf.ovpl</a> <a href="#">nmmcommload.ovpl</a> <a href="#">nmmconfigexport.ovpl</a> <a href="#">nmmconfigimport.ovpl</a> <a href="#">nmmconfigpoll.ovpl</a> <a href="#">nmmconnedit.ovpl</a> <a href="#">nmmdeleteattributes.ovpl</a> <a href="#">nmmdeleteurlaction.ovpl</a> <a href="#">nmmdisableperfspi.ovpl</a> <a href="#">nmmdiscocfg.ovpl</a> <a href="#">nmmenableperfspi.ovpl</a> <a href="#">nmmhealth.ovpl</a> <a href="#">nmmincidentcfg.ovpl</a> <a href="#">nmmldap.ovpl</a> <a href="#">nmmlicense.ovpl</a> <a href="#">nmmloadattributes.ovpl</a> <a href="#">nmmloadmib.ovpl</a> <a href="#">nmmloadnodegroups.ovpl</a> <a href="#">nmmloadseeds.ovpl</a> <a href="#">nmmmanagementmode.ovpl</a> <a href="#">nmmnetdeletenodeattrs.ovpl</a> <a href="#">nmmnetloadnodeattrs.ovpl</a> <a href="#">nmmnodedelete.ovpl</a> <a href="#">nmmnoderediscover.ovpl</a> <a href="#">nmmofficialfqdn.ovpl</a> <a href="#">nmmopcexport.ovpl</a> <a href="#">nmmprincipalconfig.ovpl</a> <a href="#">nmmresetembdb.ovpl</a> <a href="#">nmmrestore.ovpl</a> <a href="#">nmmrestoreembdb.ovpl</a> <a href="#">nmmseeddelete.ovpl</a> <a href="#">nmmsetdampenedinterval.ovpl</a> <a href="#">nmmsetdomain.ovpl</a> <a href="#">nmmsetofficialfqdn.ovpl</a> <a href="#">nmsnmpbulk.ovpl</a> <a href="#">nmsnmpget.ovpl</a> <a href="#">nmsnmpnext.ovpl</a> <a href="#">nmsnmpset.ovpl</a>	<a href="#">disco.NoVLANIndexing</a> <a href="#">disco.SkipXdpProcessing</a> <a href="#">hostnolookup.conf</a> <a href="#">ipnolookup.conf</a> <a href="#">logging.properties</a> <a href="#">lwsofmconf.xml</a> <a href="#">nmm.ports</a> <a href="#">nmm.properties</a> <a href="#">ldap.properties</a> <a href="#">nmmtrapd.conf</a> <a href="#">trapFilter.conf</a> <a href="#">UnnumberedNodeGroup.conf</a> <a href="#">UnnumberedSubnets.conf</a>

[nnmsnmpwalk.ovpl](#)  
[nnmstatuspoll.ovpl](#)  
[nnmtopodump.ovpl](#)  
[nnmtrapconfig.ovpl](#)  
[nnmtrapdload.ovpl](#)  
[nnmtrimincidents.ovpl](#)  
[nnmwhat](#)  
[ovaddobj](#)  
[ovdelobj](#)  
[ovjboss](#)  
[ovserror](#)  
[ovspmd](#)  
[ovstart](#)  
[ovstop](#)  
[pmd](#)  
[pmdmgr](#)

## Name

nmecatgets — Get localized catalog strings for NNMi

## SYNOPSIS

```
nmecatgets [ -f CATFILE -s SETNUM -m MSGNUM -d DEFAULT ]
```

## DESCRIPTION

nmecatgets is used to query a message catalog and retrieve a localized (non-English) string from that catalog file. This string can then be displayed, for example from a perl script. The caller must know the message catalog containing the desired localized strings, as well as the set-number and message-number within that file. If the message specified is not found, then the default (English) value is returned.

Note that generally this command is used internally by other NNMi or HP-provided scripts. It doesn't provide much value for other uses.

## Parameters

nmecatgets recognizes the following options.

-f *FILE*

The filename containing the localized strings to query.

-s *SETNUM*

The set number within the catalog file, generally "1".

-m *MSGNUM*

The message number within the catalog file.

-d *DEFAULT*

The default (English) string to be printed in the event that the *FILE*, *SETNUM*, and/or *MSGNUM* is not available.

## RETURN VALUE

nmecatget returns the localized string when the *FILE*, *SETNUM*, and *MSGNUM* are all valid. Otherwise it returns the *DEFAULT* (English) string provided.

## AUTHOR

nmscatgets was developed by Hewlett-Packard Company.

[Return to Reference Pages Index](#)

## Name

`nnmcluster` — start NNM cluster services

## SYNOPSIS

```
nnmcluster [ [-disable|-enable] [-display] [-startnnm|-stopnnm] [-acquire|-relinquish] [-shutdown [-force]] [-dbsync] [-halt] [-node nodename] [ [-daemon]]
```

## DESCRIPTION

`nnmcluster` starts the NNM cluster process. The NNM cluster allows an administrator to setup two systems for ensuring the availability of NNM services in the event that one system fails. When each node runs the `nnmcluster` command, they will detect each other and form a cluster. The first node to join the cluster will come up in "active" state, and starts the NNM services (via `ovstart`). The second node will detect that there is already an active node, and it will assume "standby" state. If the standby node loses connectivity with the active node (e.g. due to system shutdown or failure) then the standby node will assume active status and start NNM services.

If `nnmcluster` is called with no command-line parameters, it starts the cluster in interactive mode. Interactive mode allows the system administrator to view and modify cluster settings in an interactive session. These settings include, for example, enable or disable automatic failover, shutdown a node in the cluster, transfer NNM services from active to standby, etc.

If `nnmcluster` is called with the `-daemon` parameter, the NNM cluster is launched as a background daemon process or Windows service.

Otherwise, if called with other command-line parameters, the `nnmcluster` command will initiate the actions specified on the command-line. These actions typically affect the NNM cluster daemon process on the local node, but the `-node nodename` option will affect the NNM cluster daemon process on the node specified.

Most of the options that are available on the command-line are also available in interactive mode. For example `"-shutdown"` on the command line is the same as the `"shutdown"` command in interactive mode. Interactive mode also has some additional commands: e.g. `"help"` to display a list of available commands and `"quit"` to exit interactive mode. The `"-node nodename"` is also available interactively.

Note that only NNM cluster daemon processes are capable of starting NNM services. The interactive mode and specifying actions on the command-line are methods for affecting the behavior of a daemon process on one of the nodes in the cluster. For example, `"-acquire"` causes the daemon process on the local node (or the specified node if used with `-node` option) to acquire active state and start NNM services. Once a NNM cluster daemon process is started the only way of interacting with that daemon process is via command line or interactive mode settings, e.g. if you wish to terminate that daemon process you should use `nnmcluster -shutdown`.

When NNM is using the embedded database, the NNM cluster application synchronizes the database between the active and standby nodes. This is achieved by sending a complete database backup to the standby, followed by periodic incremental database transaction logs. The time intervals for the frequency of full backups and transaction logs are defined in the `nms-cluster.properties` file along with other cluster

For the embedded database scenario, the NNM cluster application has a startup period to allow the active node to send the database to the standby. During this startup period commands that would transfer active state to the standby node are disabled: shutdown, acquire/relinquish, etc. as they can leave the standby node in a state where it has "part" of the database and is unable to run NNM. However, once the standby node has received the database, there should be no critical times from that point forward provided both systems stay running (i.e. a restart will cause the standby node to verify that it is still in-sync with the active node).

If NNM is using the Oracle database, then no database replication is performed - the Oracle database instance is on a third machine outside of NNM control. In this scenario, the NNM cluster command still monitors the active NNM server and starts NNM on the standby server if the active fails.

## Parameters

`nnmcluster` recognizes the following options. Any unrecognized options are reported by a usage message. Options are always processed in the order specified, e.g. "`-display -disable`" is different than "`-disable -display`". All the parameters refer to the NNM cluster daemon process on the local node, unless the `-node nodename` option is used.

`-disable`

Disable automatic failover capability (automatic failover is enabled by default). The system administrator might choose to shutdown the active node briefly to perform some administrative tasks. The `-disable` parameter allows the active node to be shutdown without the standby node becoming active and starting NNM services. The same node can then be restarted in active mode using the `-acquire` followed by the `-enable` parameter.

`-enable`

Re-enable automatic failover capability after being disabled as described above.

`-display`

Connect to the cluster, query the current cluster state, and display that state to the administrator.

`-startnnm`

The active node may not be running NNM services, e.g. if those services were stopped using the `-stopnnm` option sometime earlier. The `-startnnm` option will startup those NNM services on the active node.

`-stopnnm`

Shutdown NNM services on the active node, but do not release the "active" state. This will not generate a failover event, i.e. the standby node will not assume active state.

`-acquire`

The system administrator may want to transfer NNM services from a node which is currently in the "active" state to another node which is currently in the "standby" state. Without the `-node` parameter the local system will become the new-active node. If the `-node nodename` parameter is provided, then the specified node will become the new-active node.

`-relinquish`

Allows the system administrator to give up active-state on the local (currently-active) node and cause NNM services to transfer on the current-standby node. The node being relinquished will stop NNM services and assume standby state.

`-dbsync`

Allows the system administrator to trigger an immediate database sync on active node. This option is only applicable when NNM is using the embedded database (not Oracle).

`-shutdown [-force]`

Shuts down the NNM cluster daemon process on the local node. If the `-node` option is provided, then shutdown the NNM cluster daemon process on the specified node. The NNM cluster program tries to prevent you from shutting down the cluster at times when doing so might compromise data-integrity on the standby server. For example, if the standby server is receiving a full database backup from the active server, it would be bad to have a failover event during that time. When you try to shutdown you may get a message that the cluster is in a transition state, i.e. the standby is receiving important data, so you need to allow it to complete before shutting down. If you want to force shutdown anyway, use the `"-force"` option. This will disable failover to the standby (since it is in an incomplete state), and then shutdown the local (or specified) node.

`-halt`

Shuts down the NNM cluster daemon process on all nodes in the cluster. This disables failover, then shuts down all standby nodes, and finally the active node.

`-node nodename`

Causes the action(s) specified on the command line to affect the NNM cluster daemon process on the specified *nodename*. Without this parameter the local NNM cluster daemon is affected.

`-daemon`

Starts the NNM cluster as a daemon. The command is immediately launched in the background. No other command line parameters can be specified in daemon mode.

## RETURN VALUE

`nmcluster` with command line parameters exits with the status 0 (zero) if no errors were encountered, 1 otherwise. In interactive mode the exit status is always 0.

In daemon mode the `nmcluster` command launches as a background process and the shell prompt is immediately returned. When you start a daemon process, you can monitor cluster status via the `nmcluster -display` or `ovstatus` commands. I.e. the NNM cluster will determine if/when to start NNM services (depending on active/standby status of the node in relation to other nodes in the cluster). `ovstatus` will report back a "not running" status on the standby node, but `nmcluster -display` will tell you that the node is in standby state.

## DIAGNOSTICS

`nmcluster` logs output to the NNM log directory (`/var/opt/ov/log/nm` on Unix, and `%NmDataDir%\log\nm` on Windows). Each separate thread (e.g. daemon running in parallel with interactive and/or CLI mode) has a separate log file. The most-recent running threads will always be

```
"nmcluster.0.*.log".
```

The NNM cluster internally uses an open-source technology called "JGroups". The log files for JGroups are in the same directory as above, called "jgroups.log".

## EXAMPLES

```
nmcluster -daemon  
nmcluster -display
```

The first command starts the NNM cluster as a daemon process. This command immediately returns leaving a background daemon process. That process can be queried, shutdown, etc. using interactive or CLI modes. The second command can be used (repeatedly) to monitor the status of the cluster and specifically the local daemon process to determine if it comes up in active or standby mode.

```
nmcluster -shutdown -node xyz.mycompany.com
```

Causes the NNM cluster daemon process on the specified node to shutdown. If that node is the current active node, and if automatic failover is enabled, then the NNM services will transfer to the standby node.

```
nmcluster
```

Enter interactive mode of the NNM cluster command to view and/or modify cluster parameters. Exit this program using the "exit" or "quit" commands.

```
nmcluster -acquire
```

Cause the daemon-mode NNM cluster process on the local system to become the active node. The current active is set to standby mode then NNM services will start on the local node.

```
nmcluster -disable -shutdown
```

(Perform some system administration tasks)

```
nmcluster -daemon  
nmcluster -display
```

(used to monitor when the daemon is up)

```
nmcluster -enable
```

This is a common system administration sequence to allow the administrator to temporarily shutdown the NNM cluster on the active node and then bring it back up later as active, without causing a failover event to standby. The first command will disable failover first, then shutdown the local daemon process. The system administrator can then perform some administration tasks. The second command will restart the daemon NNM cluster process. The third command can be repeated until the local daemon process is up and NNMi is running. The final command should be used to re-enable automatic failover after NNMi is running on the active node.

## AUTHOR

nmcluster was developed by Hewlett-Packard Company.

## FILES

`$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties` defines the cluster parameters; specifically the cluster name is required to uniquely name your cluster vs. other NNM clusters that might be on the same network. Other (optional) parameters can be set to define timeouts, etc.

## SEE ALSO

`ovstart(1M)`, `ovstop(1M)`.

## EXTERNAL INFLUENCES

### Environmental Variables

[Return to Reference Pages Index](#)

## Name

`nnm.envvars` — script to define the environment variables for universal pathnames in NNMi

## SYNOPSIS

Windows operating systems:

```
nnm.envvars.bat
```

UNIX operating systems:

```
nnm.envvars.sh
```

```
nnm.envvars.csh
```

## DESCRIPTION

`nnm.envvars` is a script that defines the NNMi environment variables for universal pathnames. Universal pathnames simplify the use of NNMi and other HP Software products by providing pathnames and filenames common to all operating system platforms. Universal pathnames are provided for the Windows command interpreter and UNIX shells.

On UNIX operating systems, you can modify the `.profile` or the `.login` file, so that the shell script is sourced (activated) each time you log in to your system. Alternatively, you can activate the file for an individual terminal, user, and session by sourcing the file into the current environment.

To view the universal pathnames that are defined, read the `nnm.envvars.sh` file, which resides in `/opt/OV/bin` on UNIX operating systems or `%NnmInstallDir%\bin` on Windows operating systems.

## EXAMPLES

To modify the `.profile` or `.login` file, add the appropriate line from the following examples to the file.

To source the `nnm.envvars` script, use one of the following commands.

- From a Windows command line:

```
%NnmInstallDir%\bin\nnm.envvars.bat
```

Where: `%NnmInstallDir%` is the directory where NNMi is installed. Note that the NNMi installer creates this variable as a system environment variable.

- From a Windows operating system with UNIX-style shells installed (for example, MKS Toolkit or CygWin):

Using `sh`:

```
$NnmInstallDir/bin/nnm.envvars.sh
```

Where: `$NnmInstallDir` is the directory where NNMi is installed.

Using `cs`h:

```
source $NnmInstallDir/bin/nnm.envvars.csh
```

Where: `$NnmInstallDir` is the directory where NNMi is installed.

- From UNIX operating systems:

Using `sh`, `ksh`, or `bash`:

```
. /opt/OV/bin/nnm.envvars.sh
```

Using `cs`h:

```
source /opt/OV/bin/nnm.envvars.csh
```

## AUTHOR

`nnm.envvars` was developed by Hewlett-Packard Company.

## FILES

UNIX operating systems:

`/opt/OV/bin/nnm.envvars.sh` (for `sh`, `ksh`, or `bash`)

`/opt/OV/bin/nnm.envvars.csh` (for `cs`h)

Windows operating systems:

`%NnmInstallDir%\bin\nnm.envvars.bat` (for Windows command line)

`%NnmInstallDir%\bin\nnm.envvars.csh` (for `sh`, `cs`h, or `bash`)

`%NnmInstallDir%\bin\nnm.envvars.sh` (for `sh`)

## EXTERNAL INFLUENCES

International Code Set Support: Supports single-byte and multi-byte character code sets.

[Return to Reference Pages Index](#)

## Name

`nnmprops` — Query values of NNMi properties

## SYNOPSIS

```
nnmprops [ [-l] [-q prop] [-m match] [-e expand]]
```

## DESCRIPTION

`nnmprops` is used to query the properties-values used to run NNMi processes. These properties are stored in various locations on the filesystem, which the `nnmprops` command consolidates and displays as a whole. This command could be used, for example, in other scripts that need to query and act upon the value of NNMi system properties.

## Parameters

`nnmprops` recognizes the following options.

`-l`

List out the value of all properties.

`-q PROP`

Query the specified property. This option can be repeated to query multiple properties.

`-m STRING`

Query all properties that start with the prefix *STRING*.

`-e STRING`

Expand any properties in the *STRING* with their corresponding value.

## RETURN VALUE

`nnmcluster` always exits with the status 0 (zero) if no errors were encountered, 1 otherwise.

## EXAMPLES

```
nnmprops -l
```

List all properties, and their values.

```
nnmprops -q com.hp.ov.nms.abc -q com.hp.ov.nms.xyz
```

Query the value of the "com.hp.ov.nms.abc" and "com.hp.ov.nms.xyz" properties.

```
nnmprops -m com.hp.ov.nms.foo
```

Query all properties that start with "com.hp.ov.nms.foo". For example: "com.hp.ov.nms.foo.name", "com.hp.ov.nms.foo.address", and "com.hp.ov.nms.foo.phone" would be returned.

```
nnmprops -e "The value of abc is ${com.hp.ov.nms.abc}"
```

Prints back the provided string, with the value of `${com.hp.ov.nms.abc}` expanded. Note that the "\${" and "}" are required to delineate the property name.

## AUTHOR

nnmprops was developed by Hewlett-Packard Company.

## FILES

There are several properties files used by the nnmprops program. These files are separated into two categories: HP-provided default (out-of-the-box) values, and customer-modified over-ride values. The intention is that HP can change the default properties values in future NNMi releases. However customer-modified values always over-ride the HP-provided default.

The files found in the directory hierarchy below `NnmInstallDir/misc/nnm/props` define the HP-provided default values. Customers should never modify any file below this location, as future NNMi versions will overwrite any modifications.

The files found in the directories `$NnmDataDir/shared/nnm/conf/props` and `$NnmDataDir/conf/nnm/props` define the customer-modified values, or values changed programmatically at install-time or runtime. These values are initially all commented-out copies of the HP-provided files. The customer can edit these files, uncomment and change a value, and this new value will over-ride the HP-provided default value. The difference for the two directories is that the former contains properties which are shared in a cluster (e.g. HA cluster or NNMi Application Failover cluster). The latter defines values which are not shared, i.e. where each node in the cluster may have different values for the same property.

## EXTERNALINFLUENCES

### Environmental Variables

[Return to Reference Pages Index](#)

## Name

`nnmsetcmduserpw.ovpl` — set the account credentials to be used in place of the `-u/-p` options when executing scripts as this user.

## SYNOPSIS

`nnmsetcmduserpw.ovpl`

## DESCRIPTION

You can use `nnmsetcmduserpw.ovpl` to set the NNM account credentials to be used when executing any scripts normally requiring the `-u/-p` option. The user and password values will be used when scripts are executed by this user without the `-u/-p` command line arguments. NOTE: The UNIX user root should never run this command. The "root" user does not need to provide `-u/-p` by default.

To run this command, make sure you are logged into the system as the user that will be executing the command line scripts. These values are set on a per user basis.

## Parameters

No supported parameters.

## EXAMPLES

You may want to set up account username and password values to be used when executing command line scripts normally requiring the `-u/-p` option. This can be done to avoid specifying password information on the command line.

Usage is:

```
# nnmsetcmduserpw.ovpl
```

```
WARNING: This change will affect the credentials to be used in place
of the -u/-p command line options whenever this user executes
a script requiring these. Please ensure you are logged in as
the desired user before executing this script.
Executing this script will create/edit the .nnm/nnm.properties
file in the users home directory.
Would you like to continue? [n] y
```

Thank you!

```
Please provide the user name for the account.
user: myuser
Please provide a password for the user account.
password: mypass
enter password again: mypass
```

```
User/Password values stored successfully in /home/user/.nnm/nnm.properties
```

## AUTHOR

`nnmsetcmduserpw.ovpl` was developed by Hewlett-Packard Company.

## FILES

`nnmsetcmduserpw.ovpl` resides in `$NNM_BIN` directory.

## SEE ALSO

[nnm.properties\(4\)](#).

[Return to Reference Pages Index](#)

## Name

`nnmtrapdump.ovpl` — Print traps logged in the binary trap store to the console

## SYNOPSIS

```
nnmtrapdump.ovpl [-u user] [-p password] [-t] [-from date] [-to date] [-source IP address] [-trapid Trap OID] [-last minutes] [-short] [-nodns] [-hexDump]
```

## DESCRIPTION

All incoming traps are logged in a binary trap store by the HP NNM Trap Service. `nnmtrapdump.ovpl` can be used to see the traps that have been logged. It can also be used to watch new incoming traps. When using trap tool any log or error messages are printed to the standard error. By redirecting the standard error to a different file, you can prevent these messages from getting mixed with the actual trap dump output.

## Parameters

`nnmtrapdump.ovpl` supports the following parameters:

`-u <username>`

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-p <password>`

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-t`

Use `-t` to continuously print incoming traps. This option cannot be used with the `-to` option

`-from date`

Specifies the date from which traps are printed. The date is specified in ISO 8601 standard format: `yyyy-mm-ddThh:mm:ss[+ or -]hh:mm`. This option can be used to drill down from the analytics entries in the `trapanalytics.0.0.log` file. This option cannot be used with the `-last` option

`-to date`

Specifies the date till which traps are printed. The date is specified in ISO 8601 standard format: `yyyy-mm-ddThh:mm:ss[+ or -]hh:mm`. This option can be used to drill down from the analytics entries in the `trapanalytics.0.0.log` file.

`-source IP address`

Specifies the source IP address of the traps that should be printed. Only traps from the given source

will be printed.

`-trapid Trap OID`

Specifies the trap OID of the traps that should be printed. Only traps with the given OID will be printed.

`-last minutes`

Specifies the age of the traps to be printed. The value is in minutes. Only traps are at the most this old will be printed. This option cannot be used along with the `-from` option.

`-short`

Use `-short` to print a short format of incoming traps. Only the trap OID, arrival time and source address are printed

`-nodns`

Use `-nodns` to prevent resolution of IP addresses to node names. This speeds up the printing of traps

`-hexDump`

Use `-hexDump` to print traps in hexadecimal format

## EXAMPLES

To print all the traps in the binary trap store to the console:

```
nnmtrapdump.ovpl
```

To wait in a loop and print all incoming traps:

```
nnmtrapdump.ovpl -t
```

To print traps within the 5 minute window from 9:00AM to 9:05AM MDT on July 31 2008:

```
nnmtrapdump.ovpl -from 2008-07-31T09:00:00-06:00 -to 2008-07-31T09:05:00-06:00
```

To print traps that came within the last 5 minutes and then wait for incoming traps:

```
nnmtrapdump.ovpl -last 5 -t
```

To print traps from IP address 192.168.0.1:

```
nnmtrapdump.ovpl -source 192.168.0.1
```

## FILES

`$(NNM_DB)/traps` is the directory that contains the files that constitute the trap database.

`$(NNM_LOG)/trapanalytics.0.0.log` is the analytics log file containing information about the most frequent traps and sources sending traps.

# **AUTHOR**

`nnmtrapdump.ovpl` was developed by Hewlett-Packard Company.

[Return to Reference Pages Index](#)

## Name

`nnmfindattachedswport.ovpl` — Find the switch port to which the input end node is attached

## SYNOPSIS

```
nnmfindattachedswport.ovpl [-u <user>] [-p <password>] { -i <end node file> | -n <end node> } [-o <output file>]
```

## DESCRIPTION

The `nnmfindattachedswport.ovpl` script displays the switch port that is connected to an end node. When using the `nnmfindattachedswport.ovpl` script, specify the end node as a MAC address, an IP address, or a hostname. Specify the MAC address in upper case with no leading `0x` or `0X`. You can also specify the input as a seed file, using one line per entry within the seed file.

The display consists of the following:

- The end node.
- The switch hostname.
- The interface name of the switch port that is connected to an end node.
- The VLAN name that the end node belongs to.
- The VLAN ID that the end node belongs to.
- A status code for the interface.

The status code indicates either *Success*, if NNMi successfully retrieved the end node information, or an error code. The display is in Comma Separated Value (CSV) format. If any of the values are not present then NNMi displays the value `-1` instead. NNMi indicates the completion of the script by displaying a CSV with all the values being `-1`.

NNMi initially displays a header naming each of the values in the CSV. There is an option to redirect the output to a file. The file can then be imported into Microsoft™ Excel.

## Parameters

`nnmfindattachedswport.ovpl` supports the following parameters:

`-u <user name>`

Supply an NNMi user name to use when running the script. The user could be system, an administrator or a Level 2 operator. Required unless a [nnm.properties\(4\)](#) file exists.

`-p <password>`

Supply the password for the user. Required unless a [nmm.properties\(4\)](#) file exists.

`-i <end node file>`

Use `-i` to provide an input file containing a list of end nodes. NNMi then searches for the switch port connected to each end node. The file should have only one entry per line. The value on each line could be a MAC address, an IP address or a hostname. The MAC address needs to be specified in upper case with no leading `0x` or `0X`. This parameter is required if the `-n` is not specified.

`-n <end node>`

Use the `-n` to provide an end node for the **nmmfindattachedswport.ovpl** script. The **nmmfindattachedswport.ovpl** script finds the switch port that is connected to this end node. The end node could be a MAC address, an IP address or a hostname. The MAC address needs to be specified in upper case with no leading `0x` or `0X`.

`-o <output file>`

Use `-o` to supply a file name to which the output of the script will be redirected.

## EXAMPLES

An example of an input file is given below:

```
10.45.130.2
# this is a comment line
con5.acme.com
000087D064CB
10.12.149.4
laserj.acme.com
```

An example of the display looks similar to the following:

```
EndNode,SwitchName,IfName,VLANName,VLANId,StatusCode
10.45.130.2,-1,-1,-1,-1,UNABLE_TO_LOCATE_ENTRY_IN_FDB
con5.acme.com,-1,-1,-1,-1,UNABLE_TO_LOCATE_ENTRY_IN_ARP_CACHE
000087D064CB,10.45.130.143,2/1,Network_B_IPv4,4,SUCCESS
10.12.149.4,sw1-loop0.acme.com,Fa2/21,VLAN0490,490,SUCCESS
laserj.acme.com,sw1-loop0.acme.com,Fa2/12,mpls-intercon,169,SUCCESS
-1,-1,-1,-1,-1,-1
```

## AUTHOR

`nmmfindattachedswport.ovpl` was developed by Hewlett-Packard Company.

[Return to Reference Pages Index](#)

## Name

`nnmversion.ovpl` — display the version of Network Node Manager

## SYNOPSIS

`nnmversion.ovpl`

## DESCRIPTION

`nnmversion.ovpl` can be used to display the version of Network Node Manager that is installed. It also displays the patch number of the NNM patch that is installed. If no patches are installed, it displays a message indicating that no patches are listed.

## Parameters

`nnmversion.ovpl` does not have any options.

## EXAMPLES

Print out the version of NNM and any installed patches.

`nnmversion.ovpl`

## AUTHOR

`nnmversion.ovpl` was developed by Hewlett-Packard Company.

## FILES

The following files store product version and patch information:

Windows: `data_dir\NNMVersionInfo`

UNIX: `/var/opt/OV/NNMVersionInfo`

[Return to Reference Pages Index](#)

## Name

ovjbosspath.ovpl — script to determine where jboss is installed

## SYNOPSIS

ovjbosspath.ovpl

## DESCRIPTION

ovjbosspath.ovpl is a command used by scripts to determine where jboss is used. jboss is the underlying application architecture used by ovjboss. Although ovjboss knows where jboss resides, jboss requires that certain program files (jar files) be located in underlying directories of jboss. Because other Java applications require access to these files, this provides a standard method for retrieving the base directory path.

ovjbosspath.ovpl is used to eliminate hard-coded paths in other applications.

## Parameters

None.

## EXAMPLES

On Windows with the installation in the directory C:\Program Files(x86)\HP OpenView, running C:\Program Files(x86)\HP OpenView\bin\ovjbosspath.ovpl returns the following:

```
C:/Program Files(x86)/HP OpenView/nonOV/jboss/nms
```

This enables other applications to find jar files that exist under this directory structure, such as the following:

```
C:/Program Files(x86)/HP OpenView/nonOV/jboss/nms/server/nms/lib/nms-licensing-api.jar
```

## AUTHOR

ovjbosspath.ovpl was developed by Hewlett-Packard Company.

[Return to Reference Pages Index](#)

## Name

ovjrepath.ovpl — script to determine the version of JDK to use

## SYNOPSIS

ovjrepath.ovpl

## DESCRIPTION

ovjrepath.ovpl is a command used by scripts to determine the version of the JDK to use. Given multiple products being installed on the system, there can be multiple JDK versions installed. These versions are not guaranteed to be compatible with Network Node Manager (NNM). This script encapsulates this problem by ensuring the correct JDK is being used.

NOTE: NNM replaces JDKs from time to time. This script enables other scripts to use the new JDK without being changed.

## Parameters

None.

## EXAMPLES

On Windows with the installation in the directory C:\Program Files\HP OpenView, running C:\Program Files\HP OpenView\bin\ovjrepath.ovpl returns the following:

```
C:/Program Files/HP OpenView/nonOV/jdk/b
```

This enables scripts that others are writing to use the correct JDK.

## AUTHOR

ovjrepath.ovpl was developed by Hewlett-Packard Company.

## FILES

```
$InstallDir/nonOV/jdk
```

Directory where JDKs are installed.

[Return to Reference Pages Index](#)

## Name

ovstatus — report status of NNM managed processes

## SYNOPSIS

```
ovstatus [ [-c] [-d] [-v] [managed_process_names...]]
```

## DESCRIPTION

ovstatus reports the current status of the NNM managed processes. ovstatus sends a status request (OVS\_REQ\_STATUS) to the process management process (UNIX operating system) or service (Windows operating system), ovspmd. If called with one or more *managed\_process\_name* arguments, it reports the status for the designated managed processes. If called with no arguments, it reports the status of all managed processes that have been added to the NNM startup file (SUF), including ovspmd itself.

Unlike ovstart, ovstatus does *not* start ovspmd if it is not already running.

The managed processes are configured by ovaddobj from information in Local Registration Files (see lrf(4)). A managed process is named by the first field in the LRF describing it.

## Parameters

ovstatus recognizes the option described below. The first argument that is not an option, and any succeeding arguments, are interpreted as names of managed processes for which to report status, and are passed to ovspmd in the status request.

- c  
Output one status line for each managed process.
- d  
Report the important stages in its processing, including contacting and sending the status request to ovspmd, and closing the communication channel.
- v  
Print verbose messages from managed processes. In particular, this option displays the verbose message from ovuispmd describing all current ovw sessions.

## RETURN VALUE

ovstatus normally exits with the status 0 (zero). It returns a non-zero status only if there is a system problem, such as ovspmd not running.

# DIAGNOSTICS

`ovstatus` reports certain command-line errors (in particular, too many arguments) and system errors. The messages are prefixed with `ovstatus:`, and are intended to be self-explanatory. `ovstatus` also outputs error messages received from `ovspmd`. These messages are prefixed with `ovspmd:`. `ovstatus` ignores unrecognized options.

`ovstatus` reports the known state of all `OVS_WELL_BEHAVED` and `OVS_NON_WELL_BEHAVED` processes. `OVS_DAEMON` processes run outside of `ovspmd` control. They report a PID, a state of `unknown`, and a final message of `Does not communicate with ovspmd`, as `ovspmd` cannot track these processes.

Note that `ovspmd` can process multiple requests (`ovstart`, `ovstop`, or `ovstatus`) at a time. If any of these commands is being handled, the new request will be queued by type until the previous command has completed.

## AUTHOR

`ovstatus` was developed by the Hewlett-Packard Company.

## FILES

The environment variables below represent universal pathnames that are established according to your shell and platform requirements. See the `nmn.envvars` reference page (or the UNIX manpage) for information about using environment variables for the following files:

Windows: `install_dir\bin\ovstatus`

Windows: `install_dir\bin\ovspmd`

UNIX: `$NNM_BIN/ovstatus`

UNIX: `$NNM_BIN/ovspmd`

## EXTERNAL INFLUENCES

### Environmental Variables

`$LANG` provides a default value if the internationalization variables, `LC_ALL`, `LC_CTYPE`, and `LC_MESSAGES` are unset, null, or invalid.

If `$LANG` is unset, null, or invalid, the default value of `C` (or `English_UnitedStates.1252` on Windows) is used.

`LC_ALL` (or `$LANG`) determines the locale of all other processes started by `ovspmd`.

`LC_CTYPE` determines the interpretation of text as single and/or multi-byte characters, the classification of characters as printable, and the characters matched by character class expressions in regular expressions.

LC\_MESSAGES determines the language in which messages are displayed.

## SEE ALSO

[ovstart\(1M\)](#), [ovstop\(1M\)](#), [ovaddobj\(1M\)](#), [ovdelobj\(1M\)](#), [ovspmd\(1M\)](#), [nnmcluster\(1M\)](#).

[Return to Reference Pages Index](#)

## Name

`nnmdumpevents` — dump the contents of the pre-NNM Release 8.0 event database

## SYNOPSIS

```
nnmdumpevents [ [-f fileName] [-t] [-l minutes] [-d dbPath] [-x fileName] ]
```

## DESCRIPTION

`nnmdumpevents` dumps the contents of the pre-NNM Release 8.0 event database. You can dump the entire event log, the contents of a particular stream log, or the contents of a particular correlation log. The format of the resulting file is that of the pre-NNM Release 6.0 `trapd.log` file.

`nnmdumpevents` does not dump traps. This command dumps contents of the event store associated with legacy `pmd` and would only be used for viewing NNM 6.x/7.x events forwarded to the NNMi station. `nnmtrapdump.ovpl` can be used for dumping traps. See the `nnmtrapdump.ovpl(1M)` reference page for more information.

## Parameters

`nnmdumpevents` supports the following options:

`-d dbPath`

Dumps the contents of the event database contained in the directory indicated by `dbPath`, as opposed to dumping the default database.

`-t`

Tails the output (continuously watches for new events without returning).

`-l minutes`

Dumps the contents of the event database, starting from the last few minutes instead of from the beginning of the database. This option is useful with the `-t` option.

`-f fileName`

Writes the output of the command to the named file. If you do not specify this option, the output of the command is written to standard output. The format of the output is the obsoleted `trapd.log` format.

`-x fileName`

Used with the `d` option, where the `d` option specifies the path of the directory containing `statelog` file, and the `x` option specifies the `statelog` file name to be used for dumping the events.

`-s streamName`

Dumps the contents of the named stream log. The format of the output is the obsoleted `trapd.log` format.

`-c streamName`

Dumps the contents of the correlation log associated with the `streamName` to be dumped. The event uuids of the parent and correlated child event are dumped, as are the contents of the child event.

If you do not specify `-s` or `-c`, the entire contents of the event log is dumped in `trapd.log` format.

## RETURN VALUE

If the dump of the database was successful, `nnmdumpevents` exits with a return value of 0 (zero). Otherwise, a non-zero value is returned.

## AUTHOR

`nnmdumpevents` was developed by Hewlett-Packard Company.

## FILES

The environment variables below represent universal pathnames that are established according to your shell and platform requirements. See the `nnm.envvars(1)` reference page for information on universal pathnames for your platform and shell.

`$NNM_BIN/nnmdumpevents` is the location of this executable.

## SEE ALSO

`nnmtrapdump.ovpl(1)`

[Return to Reference Pages Index](#)

## Name

`nnmsnmpnotify.ovpl` — issue an SNMP notification (Trap or Inform request)

## SYNOPSIS

```
nnmsnmpnotify.ovpl [-V version] [-C community] [-p port(default:162)] [-A] [-t timeout] [-r
retries] [-d] [-T] [-a agent_addr] [-e enterprise] node trap-oid variable type value [variable
type value]...
```

## DESCRIPTION

`nnmsnmpnotify.ovpl` sends an SNMP notification request to notify another system of an event on the local system. You can acknowledge (SNMPv2 Inform) or unacknowledge (SNMPv1 or SNMPv2 Trap) the notification. You cannot send acknowledged notifications to systems that support only SNMP Version 1.

By default, the notification is unacknowledged. An SNMP Version 1 or SNMP Version 2 Trap is sent according to the protocol version specified on the command line. In this case, `nnmsnmpnotify.ovpl` terminates immediately after sending the SNMP Trap request. There is no confirmation that the notification actually reached the destination system.

Use the `-A` option to send an acknowledged notification. `nnmsnmpnotify.ovpl` sends an SNMP Version 2 Inform request to the destination system. It then waits for the corresponding acknowledgment, and retransmits if necessary. If retransmission is required, `nnmsnmpnotify.ovpl` uses the `timeout` and `retry` values specified on the command line. If an acknowledgment is received within the specified time period, and if retransmission attempts, the notification is assured to have reached the destination system. Otherwise, `nnmsnmpnotify.ovpl` terminates without any assurance that the notification reached the destination system.

`node` can be an IP-addressable system that supports SNMP, or a target name for which an SNMP proxy configuration is defined. You can identify IP nodes by Internet address or hostname. You can supply `node` in Internet address form or host name form. If instead of a node an empty string is supplied (""), then the localhost is used as the destination.

The trap type is specified as an object identifier in the `trap_oid` command-line argument. You must identify all notifications using the object identifier form. You can supply notifications defined in the SNMPv2 MIB or in a vendor-specific SNMPv1 MIB directly to `nnmsnmpnotify.ovpl`. However, you must convert traps defined in a vendor-specific SNMPv1 MIB to object identifier form before supplying them as input to `nnmsnmpnotify.ovpl`. If an empty string ("") is specified instead of a `trap_oid` then for a SNMP Version 1 trap, the Generic trap type value is set to 6 and Specific trap type value is set to 0. For a SNMP Version 2 notification the `trap_oid` variable binding is not set..

When providing trap object identifiers to `nnmsnmpnotify.ovpl`, follow these guidelines:

- To generate a trap for the six generic SNMP traps (`coldStart`, `warmStart`, `linkDown`, `linkUp`, `authenticationFailure`, `egpNeighborLoss`), use the object identifiers defined in RFC 1907 that correspond to those traps. (For example, the `coldStart` trap has the trap OID 1.3.6.1.6.3.1.1.5.1).

- To generate a trap that is not SNMP-generic but is defined in SNMPv2 form, use the NOTIFICATION-TYPE identifier from the SNMPv2-compatible MIB.
- To generate a trap that is not SNMP-generic but defined in SNMPv1 form, you must determine the trap enterprise and specific numbers from the SNMPv1-compatible MIB. Then construct an object identifier in the form *enterprise.0.specific field*. For example, consider a vendor-specific MIB for device test. The MIB defines a trap with enterprise 1.3.6.1.4.1.11.2.17.1 and specific trap field 4. The resulting trap object identifier would be 1.3.6.1.4.1.11.2.17.1.0.4.

The data passed by `nmmsnmpnotify.ovpl` to the remote node is specified as a triple of *variable,type,value*. You must supply one or more triples as command-line arguments.

Each variable is an object instance identifier in either dotted decimal format or mnemonic string format (for example, `.1.3.6.1.4.1.11.2.17.2.1.0` or `openViewSourceId.0`).

Each *type* must be one of the following types:

INTEGER

INTEGER32

IPADDRESS

COUNTER

COUNTER32

COUNTER64 (for SNMPv2c or v3 capable remote nodes)

GAUGE

GAUGE32

OBJECTIDENTIFIER

OCTETSTRING

OCTETSTRINGASCII

OCTETSTRINGHEX

OCTETSTRINGOCTAL

OPAQUE

OPAQUEASCII

OPAQUEHEX

OPAQUEOCTAL

TIMETICKS

UNSIGNED32

For a complete description of each *type*, refer to *RFC 1155* and *RFC 1902*.

The *value* must be valid for the type specified. When using a type where a hexadecimal or octal value is needed, you must fully define each byte of the value. For example, *fff* (or *17377*) is not allowed, but *0fff* (or *017377*) is. For the type *null*, you must specify a *value* on the command line. This *value* is ignored when the request is created. *value* must not be larger than 512 bytes.

## Parameters

*-v version*

Requests that a particular version of SNMP be used to communicate with the remote node. Valid choices for version are *1*, *2c* or *3*.

*-c community*

Specifies the community string to use for authentication on the remote node.

*-p port*

Specifies the port to use in communication with the remote node.

*-t timeout*

Specifies a timeout period, in tenths of seconds, when waiting for an acknowledgment for a SNMP Version 2 Inform request. This option is only valid along with the *-A* option

*-r retries*

Specifies the number of retransmissions to attempt when no acknowledgment is received for a SNMP Version 2 Inform request. This option is only valid along with the *-A* option

*-d*

Dump ASN.1 packet trace

*-T*

Prints the OID in dotted decimal format.

*-a agent\_addr*

Overrides the local host as the source of the notification with the given agent address. *agent\_addr* must be an Internet address or Internet hostname.

*-e enterprise*

Overrides the default enterprise object identifier for the notification with the given *enterprise* value.

## EXAMPLES

The following command sends an SNMP link down Inform request to the node *v2c\_node*:

```
nmmsnmpnotify.ovpl -A -v2c v2c_node .1.3.6.1.6.3.1.1.5.3
```

The following command sends an SNMP link down trap request to the node `v1_node` with the agent address set to `agent`:

```
nnmsnmpnotify.ovpl -a agent v1_node .1.3.6.1.6.3.1.1.5.3
```

## AUTHOR

`nnmsnmpnotify.ovpl` was developed by Hewlett-Packard Company.

## FILES

The environment variable below represents a universal pathname that is established according to your shell and platform requirements:

```
$NNM_BIN/nnmsnmpnotify.ovpl
```

## SEE ALSO

[nnmsnmpwalk.ovpl\(1M\)](#), [nnmsnmpset.ovpl\(1M\)](#), [nnmsnmpbulk.ovpl\(1M\)](#).

*RFC 1155, 1157, 1212: SNMP Version 1.*

*RFC 1901 - 1908, 2576, 2578, 3416 - 3418: SNMP Version 2.*

*RFC 3411 - 3415: SNMP Version 3.*

## EXTERNAL INFLUENCES

### Environmental Variables

`$LANG` determines the language in which messages appear. If `$LANG` is not specified or is set to an empty string, a default of `C` is used instead of `$LANG`. If any internationalization variable contains an invalid setting, `nnmsnmpnotify.ovpl` behaves as if all internationalization variables are set to `C`.

### International Code Set Support

Supports single-byte and multiple-byte character code sets.

NOTE: SNMP MIB values of the type `octetstringascii` are restricted to NVT-ASCII.

[Return to Reference Pages Index](#)

## Name

nmsdbmgr — controls the NNMi embedded database, including periodic database connectivity testing

## SYNOPSIS

```
nmsdbmgr [-ovspmd] [-start] [-test] [-initnmsdb] [-stop] [-status] [-kill]
```

## DESCRIPTION

nmsdbmgr is a program that provides an interface for the `ovspmd` process to control the NNMi embedded database. This program enables `ovspmd` to start, to stop, and to check the status of the embedded database.

While running, the program tests database connectivity every five minutes and updates the status message reported to `ovspmd`, depending on the results of the connectivity test. If the database test succeeds, the message reported is `Database Available`. If the connectivity test fails, the message reported is `Data Warehouse inaccessible`. If you see the failure message, you may experience database-related problems in NNMi, such as the failure to retrieve and display the node inventory in the UI.

While it is technically possible to run this command independently of `ovspmd`, doing so is highly discouraged and could result in unexpected behavior.

NOTE: If you examine the process table while the program is executing, you see that this program spawns many child processes named `postgres` or `postgres.exe`, depending on the platform. These child processes represent the embedded database itself. Having many instances of them is normal.

## Parameters

nmsdbmgr supports the following options:

`-ovspmd`

Specifies that nmsdbmgr runs as an `ovspmd`-controlled process.

`-start`

Starts the nmsdbmgr process outside of `ovspmd` control. Use only for diagnostics.

`-test`

Verifies that the nmsdbmgr process can communicate with the database. Use only for diagnostics.

`-initnmsdb`

Creates the NNMi database. Use only for diagnostics.

`-stop`

Stops the nmsdbmgr process outside of `ovspmd` control. Use only for diagnostics.

-status

Queries the current status of the database. Use only for diagnostics.

-kill

Kills the `nmsdbmgr` process outside of `ovspmd` control. Use only for diagnostics.

## EXAMPLES

Normal usage of this command is indirect, using the `ovspmd` process.

Typical usage starts or stops the process as follows:

```
# ovstart -c nmsdbmgr
# ovstop -c nmsdbmgr
# ovstatus -c nmsdbmgr
```

## AUTHOR

`nmsdbmgr` was developed by Hewlett-Packard Company.

## FILES

`$NON_OV_DIR/Postgres`

`$DATA_DIR/shared/nm/databases/Postgres`

## SEE ALSO

`ovspmd(1M)`, `ovstart(1M)`, `ovstop(1M)`, `ovstatus(1M)`.

[Return to Reference Pages Index](#)

## Name

nnmaction — wrapper around the HP NNM Action Server

## SYNOPSIS

nnmaction

## DESCRIPTION

nnmaction is a process that is managed by the `ovspmd`. The properties file (`$NNM_DATA/shared/nnm/conf/props/nnmaction.properties`) can be used to pass arguments to the Action Server.

This command should be never be executed directly. It will be started and managed by `ovspmd`. Please use the commands `ovstart` and `ovstop` to start and stop it. Please use the command `ovstatus` to find it's status.

You must be logged on as root/administrator to run this command.

## AUTHOR

nnmaction was developed by Hewlett-Packard Company.

## FILES

`$NNM_DATA/shared/nnm/conf/props/nnmaction.properties`

Parameter file used by Action Server

## SEE ALSO

`ovspmd(1M)` `ovstart(1M)` `ovstop(1M)` `ovstatus(1M)`

[Return to Reference Pages Index](#)

## Name

`nnmbackup.ovpl` — script to perform a complete or scoped backup of NNMi data and files

## SYNOPSIS

```
nnmbackup.ovpl [-?|-h|-help] [-type (online|offline)] [-scope (config|topology|events|all)] [-force] [-archive] -target <directory>
```

## DESCRIPTION

`nnmbackup.ovpl` serves as the main backup script for NNMi. For installations with an embedded database, the script determines the directories and tables that should be considered for the backup operation, using the `$NNM_DATA/shared/nnm/backup.properties` file. The `$NNM_DATA/shared/nnm/backup.properties` file also defines files and directories that require extra processing during the restore phase. The command takes various arguments to determine things like backup scope (`config`, `topology`, `events`, `all`), backup location, and backup type (`online` or `offline`).

Before running this command, make sure you have adequate storage space in the target directory. In general, if you have enough space to store the contents of the NNMi installation and `$NNM_DATA` directories, you have adequate storage space; check the following:

- `/opt/OV` (UNIX)
- `$NnmInstallDir` (Windows)

If you installed with the embedded database option, the embedded database data storage can be found in the `$NNM_DATA/shared/data/nnm/databases/Postgres` directory. You can compress the files after backup.

The target directory contains all of the files applicable for the backup options you have selected, or a single tar file if you use the `-archive` option. Each backup operation stores files in a parent directory called `nnm-bak-<TIMESTAMP>` inside of target directory. Any statements executed in the database while the backup is running are included in the backup.

Files that require extra processing during the restore phase are stored with their full paths under `target_directory/special_files/handling_routine`. It is in the restore phase that files are selected for exclusion, restoration, or merge (see `nnmrestore.ovpl`). The backup is always populated with the data necessary to perform a restore operation. You must be logged in as root on UNIX systems, or as administrator on Windows systems to run this command.

## Caution

Database backups performed by this command only apply to the embedded database. If you chose a different database at install time, your table data is not backed up using this command. File-system backups work regardless of the database type. For details about how to back up if you select a different database at install time, refer to the *NNMi Deployment Guide* (available at <http://h20230.www2.hp.com/selfsolve/manuals>).

# Parameters

`nnmbackup.ovpl` supports the following options:

`-type (online|offline)`

Determines the type of backup to be performed. If you specify `online`, NNMi must be running when you perform the backup (`nmsdbmgr` is the only required process). If you specify `offline`, NNMi must be completely stopped when you perform the backup.

`-scope (config|topology|events|all)`

Determines the scope of the backup operation. There are two types of data that are backed up: files in the file system and tables in the database. Scope for files in the file system is always applicable, regardless of the type of backup type you have chosen (see `-type`). However, scope for tables in the database is applicable only when you run an online backup (`-type online`). For offline scoped backups, you get the entire contents of the database, not just the scope you have requested. For this reason, HP recommends that you do not define scope when doing offline backups (default is `all`). The scopes available are `config`, `topology`, `events`, and `all`. Each scope includes all data and files from the previous scope (`all` → `events` → `topology` → `config`). The `$(NNM_DATA)/shared/nnm/backup.properties` file contains a list of the files and tables backed up at each scope.

`-force`

If you provide this option, the command starts and stops NNMi based on the type of backup you requested. For online backups, if NNMi is not running, the script starts the `nmsdbmgr` process (required for backups). For offline backups, if NNMi is running, the command stops all NNMi processes.

`-archive`

If you provide this option, the backup files are stored in a tar file in the target directory.

`-target <directory>`

Specifies the output directory where the backup files are stored. The command creates a parent directory named `nnm-bak-<TIMESTAMP>` inside the target directory where all backup files are stored. If the `archive` option is present, a temporary directory `nnm-bak-<TIMESTAMP>` is created and replaced by a tar file of the same name.

`-?|-h|-help`

Display command usage.

## EXAMPLES

To perform a backup using the `online` and the `config` options; for example, because you want to save a configuration before discovering your network, but not the discovery results:

```
#!/nnmbackup.ovpl -type online -scope config -target /tmp/bak/config
```

To perform a backup using the `online`, the `events` and the `force` options; for example because you want to save your configuration, discovered topology, and event data without bringing the application down:

```
#!/nnmbackup.ovpl -type online -scope events -target /tmp/bak/evt
```

To perform a backup using the `offline` and `full` options; for example, a regularly scheduled backup job:

```
#./nnmbackup.ovpl -force -type offline -scope all -target /tmp/bak/all
```

## AUTHOR

`nnmbackup.ovpl` was developed by Hewlett-Packard Company.

## FILES

`$NNM_DATA/shared/nm/backup.properties`

## SEE ALSO

`nnmrestore.ovpl(1)`.

[Return to Reference Pages Index](#)

## Name

`nnmbackupembdb.ovpl` — make a full backup of the NNMi embedded database

## SYNOPSIS

```
nnmbackupembdb.ovpl [-?|-h|-help] -target <target directory> [-force]
```

## DESCRIPTION

Use the `nnmbackupembdb.ovpl` command to make a full backup of the NNMi embedded database. The contents of the backup are uncompressed; they are stored in the file you specified. This file is required to restore using the `nnmrestoreembdb.ovpl` command.

Before running the `nnmbackupembdb.ovpl` command, make sure you have adequate storage space in the target directory. You can find the embedded database data storage in the `$(DATA_DIR)/shared/data/nnm/databases/Postgres` directory. Check the size of this directory to make sure your target directory has sufficient space to store the backup file. You can compress the file after backup if necessary.

The backup file is created in the target directory and contains all data stored in the database at the start of the backup operation. Any statements executed in the database while the backup is running are not included in the backup.

You can run this command while NNMi is running; however, you may experience temporary performance degradation. At a minimum, the `nmsdbmgr` process must be running for the command to be successful. If you use the `-force` option, the command starts `nmsdbmgr` (if it is not running), and suppresses any interactive messages.

You must be logged in as root on UNIX systems, or as administrator on Windows systems to run this command.

## Parameters

`-target <directory>`

Directory name in which to store the backup output. This directory must exist.

`-force`

If you provide this option, the command starts the `nmsdbmgr` process if it is not currently running.

`-?|-h|-help`

Display command usage.

## EXAMPLES

You can use this script to run an "on-demand" backup. Or you can include it as a task in regularly scheduled backup scripts. Enter the following:

```
# nnmbakupembdb.ovpl -target /backups/nnm
```

You'll see these messages:

```
WARNING: Running this command while NNM is running, while allowed, will cause
         temporary performance problems. At a minimum, please make sure the
         nmsdbmgr process is running (ovstart nmsdbmgr).
```

```
Are you sure you want to run a full database backup now? [n] y
```

```
Ok, performing full embedded database backup...
```

```
NNM embedded database successfully backed up to /backups/nnm/nm-bak.2009092906.pgd.
#
```

## AUTHOR

nnmbakupembdb.ovpl was developed by Hewlett-Packard Company.

## FILES

nnmbakupembdb.ovpl resides in \$NNM\_BIN directory.

## SEE ALSO

ovstart(1M), ovstop(1M), ovstatus(1M), nmsdbmgr(1M), nnmrestoreembdb.ovpl(1M).

[Return to Reference Pages Index](#)

## Name

`nnmcertmerge.ovpl` — Automation of trust and key store merges into NNMi certificate stores

## SYNOPSIS

```
nnmcertmerge.ovpl [-?|-h|-help] [-keystore <file> -truststore <file>][[-directory <directory>]]
```

## DESCRIPTION

Use `nnmcertmerge.ovpl` to automate certificate store merges into the `nnm.keystore` and `nnm.truststore`. This script greatly simplifies the task of merging all certs when NNMi is run in GeoDiverse, HA, or AppFailover environments.

You must be logged in as `root` on UNIX systems or as `administrator` on Windows systems to run this command.

## Parameters

`nnmcertmerge.ovpl` supports the following options:

`-keystore <file>`

If you provide this option the target file is merged into the `nnm.keystore`. This option can be provide at the same time as `-truststore`

`-truststore <file>`

If you provide this option the target file is merged into the `nnm.truststore`. This option can be provide at the same time as `-keystore`

`-directory <directory>`

If you provide this option all files in the target directory are handled in the following manner. All files ending in `.keystore` are merged into the `nnm.keystore`. All files ending in `.truststore` are merged into the `nnm.truststore`. This option must be provided by itself.

`-?|-h|-help`

Display command usage.

## EXAMPLES

Merge a keystore with NNMi:

```
nnmcertmerge.ovpl -keystore /tmp/hostA.keystore
```

**Merge a truststore with NNMi:**

```
nmmcertmerge.ovpl -truststore /tmp/hostA.truststore
```

**Merge a keystore and truststore with NNMi:**

```
nmmcertmerge.ovpl -keystore /tmp/hostA.keystore -truststore /tmp/hostA.truststore
```

**Merge a set of keystores and truststores with NNMi:**

```
nmmcertmerge.ovpl -directory /tmp/AppFailoverHosts/
```

## **AUTHOR**

nmmcertmerge.ovpl was developed by Hewlett-Packard Company.

## **FILES**

nmmcertmerge.ovpl resides in \$NNM\_BIN directory.

[Return to Reference Pages Index](#)

## Name

`nnmchangedbpw.ovpl` — change the username and password used to authenticate with the NNMi database

## SYNOPSIS

`nnmchangedbpw.ovpl`

## DESCRIPTION

Use `nnmchangedbpw.ovpl` to change the username and password that NNMi uses to connect to the database. This command is useful when database passwords expire, or if you need to change the database username. Before running this command, use the `ovstop` command to stop NNMi.

### Note

To avoid adverse behavior after changing the database username and password, you must stop NNMi before running `nnmchangedbpw.ovpl`. You can run this command while NNMi is running, but any new database connections created by NNMi fail during authentication.

You must be logged in as `root` on UNIX systems or as `administrator` on Windows systems to run this command.

## Parameters

No supported parameters.

## EXAMPLES

You can run this command if your organization requires frequent password changes for the NNMi database, or to change the username NNMi uses to connect to the database.

You'll see these messages:

```
# nnmchangedbpw.ovpl
WARNING: Please make sure NNM has been stopped prior to running this tool.
         Failure to stop NNM could result in unexpected database failures.
Have you stopped NNM (ovstop)? [n] y
Thank you!

Please provide a database user account name.
user: mydbuser

Please provide the password for database user account.
password: mynewpw
enter password again: mynewpw

User/Password changed successfully!
#
```

## AUTHOR

`nnmchangedbpw.ovpl` was developed by Hewlett-Packard Company.

## FILES

`nnmchangedbpw.ovpl` resides in `$NNM_BIN` directory

## SEE ALSO

`ovstart(1M)`, `ovstop(1M)`, `ovstatus(1M)`.

[Return to Reference Pages Index](#)

## Name

`nnmchangeembdbpw.ovpl` — change the password used to authenticate with the NNMi embedded database

## SYNOPSIS

```
nnmchangeembdbpw.ovpl
```

## DESCRIPTION

Use `nnmchangeembdbpw.ovpl` to change the database password that NNMi uses to connect to the embedded database. This command is useful if you installed NNMi with the embedded database option. You should use this command only if you want to change the default password that is created for the embedded database. If you do not run this command, NNMi functions normally.

Before running this command, do the following:

- Enter the `ovstop` command to stop NNMi.
- Enter the `ovstart nmsdbmgr` command to start the `nmsdbmgr` process.

### Note

If the `nmsdbmgr` process is not running, the `nnmchangeembdbpw.ovpl` command fails.

When `nnmchangeembdbpw.ovpl` has completed, the embedded database password is changed to the value supplied through the interactive prompts, and the NNMi server is reconfigured to use the new password to connect to the database.

You must be logged in as `root` on UNIX systems or as `administrator` on Windows systems to run this command.

## Parameters

No supported parameters.

## EXAMPLES

You can run this command if your organization requires frequent password changes for the NNMi embedded database, or to change the username NNMi uses to connect to the embedded database.

You'll see these messages:

```
# nnmchangeembdbpw.ovpl
```

```
WARNING: Please make sure NNM has been stopped prior to running this tool.  
Failure to stop NNM could result in unexpected database failures.
```

After stopping NNM, run `ovstart nmsdbmgr` to start only the database.

```
Have you stopped NNM (ovstop)? [n] y
Thank you!
```

```
Please provide a new password for the embedded database user account.
password: mynewpw
enter password again: mynewpw
```

```
Password changed successfully!
#
```

## AUTHOR

`nnmchangeembdbpw.ovpl` was developed by Hewlett-Packard Company.

## FILES

`nnmchangeembdbpw.ovpl` resides in `$NNM_BIN` directory.

## SEE ALSO

`ovstart(1M)`, `ovstop(1M)`, `ovstatus(1M)`, `nmsdbmgr(1M)`.

[Return to Reference Pages Index](#)

## Name

`nnmchangesyspw.ovpl` — change the password for the system account that is normally set at install time

## SYNOPSIS

`nnmchangesyspw.ovpl`

## DESCRIPTION

You can use `nnmchangesyspw.ovpl` to change the NNM system password. The system password is normally set during installation and used for recovery purposes. You should use this command only if you want to reset the system password that was specified at install time.

To run this command, stop NNM by using the `ovstop` command. After the command has been executed, restart NNM by using the `ovstart` command. This will ensure that the new value for the password is valid immediately.

To run this command, you must be logged in as root/administrator.

## Parameters

No supported parameters.

## EXAMPLES

You may want to change the system password if you have deleted all other user accounts with administrator privileges, and don't remember the system password value set at install time.

Usage is simple:

```
# nnmchangesyspw.ovpl
WARNING: This change may not take affect immediately unless NNM is
         restarted. Please run ovstop before executing this script,
         and ovstart after execution to ensure the change is immediate.
Would you like to continue? [n] y
Thank you!

Please provide a new password for the system user account.
password: mynewpw
enter password again: mynewpw

System password changed successfully

#
```

## AUTHOR

`nnmchangesyspw.ovpl` was developed by Hewlett-Packard Company.

## FILES

`nnmchangesyspw.ovpl` resides in `$NNM_BIN` directory

## SEE ALSO

[ovstart\(1M\)](#), [ovstop\(1M\)](#), [ovstatus\(1M\)](#).

[Return to Reference Pages Index](#)

## Name

`nnmcommconf.ovpl` — display communication configuration information

## SYNOPSIS

```
nnmcommconf.ovpl [-u username ] [-p password] [-jndiHost host name] [-jndiPort port Default is 1099] -  
proto <icmp | snmp> -host <hostname>
```

## DESCRIPTION

If you often run NNM command line tools it is recommended that you create an `nnm.properties` file containing your username and password. This will allow you to execute many of the commands without entering your password. This file should be placed in an `.nnm` subdirectory under your home directory, which will usually be `drive:\Documents and Settings\username\.nnm\` (Windows) or `~/\.nnm/` (UNIX).

`nnmcommconf.ovpl` is a tool that communicates with the NNM system to show how the system will attempt to communicate with a given host. It can show information based on either SNMP or ICMP.

## Parameters

`nnmcommconf.ovpl` recognizes the following options.

`-proto`

Protocol: SNMP or ICMP

`-host`

The name of the host about which you plan to retrieve information.

`-u <username>`

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-p <password>`

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-jndiHost`

The server JNDI host; default is `localhost`.

`-jndiPort`

The server JNDI port; default is 1099.

## EXAMPLES

```
nnmcommconf.ovpl -username foo -password bar  
-proto icmp -host baz
```

By running the above command, you may get the information displayed below.

```
address      =15.2.121.254  
  
timeout      =2000  
  
address      =15.2.121.254  
  
retries      =1  
  
enabled      =true  
  
region name  =default
```

```
nnmcommconf.ovpl -username foo -password bar  
-proto snmp -host baz
```

By running the above command, you may get the information displayed below.

```
name         =baz  
  
address      =15.2.121.254  
  
addressForced =false  
  
getCommunity =nnmpublic  
  
timeout      =5000  
  
retries      =1  
  
port         =161  
  
enabled      =true  
  
region name  =default
```

## AUTHOR

nnmcommconf.ovpl was developed by Hewlett-Packard Company.

## FILES

`nnmcommconf.ovpl` resides in the following directory:

Windows: `install_dir\bin\nnmcommconf.ovpl`

UNIX: `/opt/bin/nnmcommconf.ovpl`

## SEE ALSO

[nnm.properties](#)(4), `nnmcommload.ovpl`(1M).

[Return to Reference Pages Index](#)

## Name

nnmcommload.ovpl — load communication settings from a comma separated file

## SYNOPSIS

```
nnmcommload.ovpl [-u username ] [-p password] [-jndiHost host name] [-jndiPort port Default is 1099] -  
file <filepath | filename>
```

## DESCRIPTION

If you often run NNM command line tools it is recommended that you create an nnm.properties file containing your username and password. This will allow you to execute many of the commands without entering your password. This file should be placed in an ".nnm" subdirectory under your home directory, which will usually be "*drive*:\Documents and Settings\*username*\.nnm\" (Windows) or "~/nnm/" (UNIX).

You can import communication settings for a group of devices in bulk by using the nnmcommload.ovpl command line tool. This is useful when your community strings are managed by a change control mechanism. You can bulk insert the assignments into NNM. Each assignment shows up as an individual entry in the Specific Node Settings tab of the Communication Configuration dialog.

Hostnames that are specified as IP addresses will not be resolved to fully qualified names. Hostnames specified as names will be resolved to fully qualified names via DNS to validate, so it will take some time to work through large import files. For files with more than 500 lines, the entries will be saved to the database in batches of 500. Once 500 lines from the import file have been read, the SNMP configuration entries for those lines will be resolved based on any existing SNMP region or default settings and saved to the database.

To perform the import, create a text file that includes the following information. Create one line for each device. Within each line, place the information in the following order, separating each value with a comma. Comment lines start with a # character. The data will be interpreted strictly by position within the line, so a comma must be specified as a placeholder for non-specified values. Embedded commas are allowed by quoting the string. For example: "comm,string"

- Target node name or IP address (required)
- Read community string (optional)
- Management address (optional)
- Write community string (optional)
- Timeout in milliseconds (optional)
- Number of retries (optional)
- Port (optional)
- Proxy address (optional)

- Proxy port (optional)
- User name (SNMP V3 optional)
- Context name (SNMP V3 optional)
- Authentication protocol (SNMP V3 optional - HMAC\_MD5\_96 or HMAC\_SHA\_1)
- Authentication password (SNMP V3 optional)
- Privacy protocol (SNMP V3 optional - DES|3DES|AES128|AES192|AES256)
- Privacy password (SNMP V3 optional)
- Preferred SNMP version (optional - 1|2|3)
- Enable SNMP Communication flag (optional - true|false)
- Enable SNMP Address Rediscovery flag (optional - true|false)
- Enable SNMP GetBulk flag (optional - true|false)

For example, all of the following entries are considered valid entries:

hostname

hostname,

hostname,,

hostname,,public

hostname,,15.2.23.33,,1000,2,161

node1,community,10.3.7.96,writcommunity

node2, community,10.3.7.95 (the space character before community will be removed)

15.2.23.34,community,15.2.23.88

15.2.23.34,community,15.2.23.88,writcommunity,2000,2,161,10.56.22.199,162

## Parameters

nnmcommload.ovpl recognizes the following options.

-u <username>

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

-p <password>

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

-jndiHost (optional)

The hostname where the jboss application server is running, defaulting to `localhost` if not specified.

`-jndiPort` (optional)

The jboss application server port, defaulting to 1099 if not specified.

`-file` (required)

A filename or full pathname to a file created with the format described above which contains the data to be loaded.

## EXAMPLES

```
nnmcommload.ovpl -user joe -password secret import.txt
```

```
nnmcommload.ovpl -user joe -password secret C:\temp\import.txt
```

```
nnmcommload.ovpl -user joe -password secret /tmp/import.txt
```

```
nnmcommload.ovpl -user joe -password secret -jndiHost myserver -jndiPort 1117 secret import.txt
```

## AUTHOR

`nnmcommload.ovpl` was developed by Hewlett-Packard Company.

## FILES

`nnmcommload.ovpl` resides in the following directory:

Windows: `install_dir\bin\nnmcommload.ovpl`

UNIX: `/opt/bin/nnmcommload.ovpl`

## SEE ALSO

[nnm.properties\(4\)](#), [nnmcommconf.ovpl\(1M\)](#).

[Return to Reference Pages Index](#)

## Name

`nnmconfigexport.ovpl` — export configuration to a file(s) which can be imported on another system

## SYNOPSIS

```
nnmconfigexport.ovpl -? | -c <configuration>[,configuration...] [-a <author_key>] [-u <username> -p <password>] [-x <file_prefix>] [-f <output file or directory>]
```

## DESCRIPTION

`nnmconfigexport.ovpl` is a Perl script that enables you to export the customized configuration to stdout, or save it in a file.

## Parameters

`nnmconfigexport.ovpl` supports the following options:

`-?`

Prints the usage statement.

`-u <username>`

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-p <password>`

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-c <configuration>[,configuration...]`

Exports the XML schema for the specified configuration. Use a comma separated list for multiple configurations (note that if multiple configurations are present, the `-f` option must specify a directory).

Valid configurations:

`account`

Exports User Accounts, User Roles and User Principals.

`author`

Exports Authors. Can be optionally filtered using the `-a` argument.

`customCorrelation`

Exports Custom Correlation Configuration. Can be optionally filtered using the `-a` argument.

`comm`

Exports Communication Configuration. SNMPv3 Communication Configuration and Device Credentials will not be exported. The encryption algorithms that are used with these data rely on an internal key specific to the NNMi installation. It is not possible to import these data so they are excluded from the export.

`custpoll`

Exports Custom Poller Configuration.

`device`

Exports Device Profiles. Can be optionally filtered using the `-a` argument.

`disco`

Exports Discovery Configuration (does not include seeds).

`discoseed`

Exports discovery seeds.

`ifgroup`

Exports Interface Groups.

`iftype`

Exports Interface Types (IfTypes). Can be optionally filtered using the `-a` argument.

`incident`

Exports Incident Configuration. Can be optionally filtered using the `-a` argument.

`menu`

Exports Menus. Can be optionally filtered using the `-a` argument.

`menuitem`

Exports all Menu Items that are configured for the Actions menu. When the `-a` argument is supplied, the output will also contain the related parent menus/sub menus.

`mibexpr`

Exports MIB expressions. Can be optionally filtered using the `-a` argument.

`monitoring`

Exports Monitoring Configuration.

`nodegroup`

Exports Node Groups.

`ngmap`

Exports Node Group Maps. Node host names must match between both machines for node

coordinates to import successfully.

rams

Exports route analytics management server(s) configuration

NOTE: The `Query Password` field of a RAMS configuration will only be valid when imported into the same NNMi installation on the same system. If imported into a different system, the `Query Password` will need to be re-entered.

station

Exports NNM 6.x/7.x Management Stations.

status

Exports the Node Group Status Configuration.

ui

Exports User Interface Settings.

all

Exports all of the available configuration areas - output must be to a directory for this option.

`-a <author_key>`

Export only the configuration items created by author with key `author_key` in a special XML format used for incremental import. `nnmconfigimport.ovpl` will detect this XML format automatically -- no special option in `nnmconfigimport.ovpl` is required. This option is only available for configurations `author`, `customCorrelation`, `device`, `incident`, `menu`, and `menuitem`. Available author keys can be found doing an export of authors, see examples below.

`-f <output file or directory>`

Saves the output to the specified file or directory.

`-x <file_prefix>`

A filename prefix that is used to name files when the output specified is a directory. Files will be named `<prefix>-<area>.xml`.

## EXAMPLES

```
nnmconfigexport.ovpl -u myusername -p myadminpassword -c comm
```

Exports the communication configuration to stdout.

```
nnmconfigexport.ovpl -u myusername -p myadminpassword -c comm,disco -f /tmp -x my
```

Exports the communication and discovery configurations to files named `/tmp/my-comm.xml` and `/tmp/my-disco.xml`.

```
nnmconfigexport.ovpl -u myusername -p myadminpassword -c author
```

Exports all authors with author key and label to stdout.

```
nnmconfigexport.ovpl -u myusername -p myadminpassword -c menuitem -a  
com.mycorp.nnm.author -f /tmp/mycorpmenuitems.xml
```

Exports the menuitem configuration created by author with key `com.mycorp.nnm.author` to the `/tmp/mycorpmenuitems.xml` file.

## AUTHOR

`nnmconfigexport.ovpl` was developed by Hewlett-Packard Company.

## SEE ALSO

[nnmconfigimport.ovpl\(1M\)](#), [nnm.properties\(4\)](#)

[Return to Reference Pages Index](#)

## Name

`nnmconfigimport.ovpl` — import the `nnmconfigexport.ovpl` XML output to the NNMi database

## SYNOPSIS

```
nnmconfigimport.ovpl -? | [-u <username> -p <password>] -f <input file or directory> [-x <file prefix>] [-memory <number of megabytes>] [-timeout <time in seconds>]
```

## DESCRIPTION

`nnmconfigimport.ovpl` is a Perl script that enables you to import the output of `nnmconfigexport.ovpl` into the NNMi database.

**NOTE:** When performing an action import, you must restart the UI (sign out/sign in) for the changes to take effect.

## Parameters

`nnmconfigimport.ovpl` supports the following options:

`-?`

Prints the usage statement.

`-u <username>`

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-p <password>`

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-f <input file or directory>`

Imports the configuration XML file, or if a directory is specified, all of the files in that directory.

`-x <file prefix>`

The file prefix used to filter files to be imported when the `-f` option specifies a directory. Files within the specified directory that are named `<prefix>-*` will be imported, sorted by dependencies.

`-memory <memory in megabytes>`

The amount of memory available for the `nnmconfigimport.ovpl` script to perform work. The default is 512 megabytes and larger import files may require this value be set to 1024 or 2048. This option is not presented in the usage message since it is used by `nnmconfigimport.ovpl` and not passed to the actual import tool.

`-timeout <time in seconds>`

The amount of time available for the import of a particular file to complete. Some import types such as incidents might require a larger timeout value based upon the amount of data. The default is 1800 (30 minutes). This option is not presented in the usage message since it is used by `nnmconfigimport.ovpl` and not passed to the actual import tool.

## NOTES

By default `nnmconfigimport.ovpl` replaces the existing configuration.

In order to add configuration elements `nnmconfigexport.ovpl` must be run with a provided `author_key`.

`nnmconfigimport.ovpl` will automatically detect if `nnmconfigexport.ovpl` was executed with a provided `author_key`, and add configuration entries instead of replacing them.

The `Query Password` field of a RAMS configuration will only be valid when imported into the same NNMi installation on the same system. If imported into a different system, the `Query Password` will need to be re-entered.

Do not edit the `nnmconfigexport.ovpl` output files before importing.

## EXAMPLES

```
nnmconfigimport.ovpl -u system -p openview -f /tmp/nnmconfig.xml
```

Imports the customized configuration in the `/tmp/nnmconfig.xml` file to the NNMi database. (You must provide an NNMi username and password. In this case, the username is `system` and the password is `openview`.)

## AUTHOR

`nnmconfigimport.ovpl` was developed by Hewlett-Packard Company.

## SEE ALSO

[nnmconfigexport.ovpl\(1M\)](#), [nnm.properties\(4\)](#).

[Return to Reference Pages Index](#)

## Name

`nnmconfigpoll.ovpl` — poll a node for discovery information

## SYNOPSIS

```
nnmconfigpoll.ovpl [-v] [-t timeout in secs] [-u <username> -p <password>] [-jndiHost <hostName>
Default: localhost] [-jndiPort <port> Default: 1099] node
```

## DESCRIPTION

`nnmconfigpoll.ovpl` sends a request to the discovery service to poll the node for discovery information. The node must exist in the discovered topology and can be the node name in the topology or an IP address associated with the node.

NNM displays the status messages for the layer 3 discovery information on the console as the discovery service polls the device. As a result of executing the `nnmconfigpoll.ovpl` command, a layer 2 connectivity analysis is started for the node.

Note that this command polls for discovery information. Status information should be polled using the `nnmstatuspoll.ovpl` command.

## Parameters

The `nnmconfigpoll.ovpl` command recognizes the following parameters and options:

`-v`

Display verbose information about the discovery poll.

`-t <timeout in secs>`

Client waits till given timeout in sec.

`-u <username>`

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-p <password>`

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-jndiHost <serverName>`

The server JNDI host; default is `localhost`.

`-jndiPort <port>`

The server JNDI port; default is 1099.

## EXAMPLES

The following examples show how to use the `nnmconfigpoll.ovpl` command to poll a node several different ways.

Poll a node using its node name.

```
nnmconfigpoll.ovpl -u username -p password thisnode
```

Poll a node using its fully qualified node name.

```
nnmconfigpoll.ovpl thisnode.x.y.z
```

Poll a node using the IP address of the node.

```
nnmconfigpoll.ovpl 10.97.247.129
```

## AUTHOR

`nnmconfigpoll.ovpl` was developed by Hewlett-Packard Company.

## FILES

Windows: `install_dir\bin\nnmconfigpoll.ovpl`

UNIX: `$OVBIN/nnmconfigpoll.ovpl`

## SEE ALSO

`nnmstatuspoll.ovpl(1M)`.

[Return to Reference Pages Index](#)

## Name

nnmconnect.ovpl — make corrections to the L2 connection topology. Users can add and delete connections

## SYNOPSIS

```
nnmconnect.ovpl -f corrections file -t [add|delete] [-help] [-u <username> -p <password>] [-jndiHost <hostName> Default: localhost] [-jndiPort <port> Default: 1099]
```

## DESCRIPTION

Due to a variety of factors, NNM L2 connection topology discovery can contain inaccuracies. The connection editor tool provides a way for the user to add or delete connections. The corrections-file is created by the administrator and must follow this structured XML format:

```
<connectionedits>
  <connection>
    <operation>add or delete</operation>
    <node>name, long name or IP address</node>
    <interface>ifName, ifAlias, ifDescr or ifIndex</interface>
    <node>name, long name or IP address</node>
    <interface>ifName, ifAlias, ifDescr or ifIndex</interface>
  </connection>
</connectionedits>
```

Where:

*operation* identifies whether the connection is to be added or deleted.

*node* is identified by its short name, long name (i.e., DNS name) or IP address.

*interface* is identified in order by ifIndex, ifName, ifDescr, or ifAlias. This value must be unique. Note that using ifIndex is discouraged due to the interface renumbering feature supported by some devices. For non-SNMP nodes, ifAlias or ifDescr are recommended.

For each connection element, there must be at least two nodes and at least two interfaces. The number of nodes and number of interfaces must be equal. Each node/interface pair is known as an endpoint, so a single connection element can have two or more endpoints specified. Multiple connection elements are allowed within a corrections file.

When adding a connection, each endpoint will be removed from any existing connection of which it may be a member, and added to the new connection. If there are more than two endpoints specified in the connection element, the connection will appear on maps as a shared-media connection symbol. If the connection element specifies a connection which already exists in the NNMi database, then nothing will be changed.

When deleting a connection, nothing will be changed unless a connection with the same set of endpoints already exists in the NNMi database. In that case, all of the specified endpoints will be left in a disconnected state.

# Parameters

The `nnmconnect.ovpl` command recognizes the following parameters and options:

`-f corrections file`

Specifies the name of the file that contains the formatted connection add and delete directives.

`-t [add | delete]`

Generates a template file that can be used to create a corrections file. Specifying *add* will create an add operation template while specifying *delete* will generate a delete operation template.

`-help`

This option displays the command usage information.

`-u <username>`

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-p <password>`

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-jndiHost <serverName>`

The server JNDI host; default is `localhost`.

`-jndiPort <port>`

The server JNDI port; default is `1099`.

## EXAMPLES

Suppose that NNM connection discovery was unable to find the L2 connection between two switches manufactured by different vendors. To remedy this, go to each device and get the node names and interface names that need to be connected. Next, create the add template file, `mychg.xml`, passing the `-t` option of the `nnmconnect.ovpl` command. Next, edit the file, filling in the node and interface information. Save your changes in the `mychg.xml` file.

The following is an example of the add template file:

```
<connectionedits>
  <connection>
    <operation>add</operation>
    <node>nodeA.x.y.z</node>
    <interface>fa/09</interface>
    <node>nodeB.x.y.z</node>
    <interface>fa/05</interface>
  </connection>
</connectionedits>
```

Finally, run the command passing the `mychg.xml` with the `-f` option.

```
nnmconnect.ovpl -f mychg.xml
```

Suppose that NNM connection discovery creates an L2 connection where none should exist. To remedy this, get the node names and interfaces involved in the incorrect connection. Then create the delete template file, passing the `t` option of the `nnmconnect.ovpl` command. Next, edit the file by filling in the node and interface information. Save your changes in `mychg.xml` file.

The following is an example of the delete template file:

```
<connectionedits>
  <connection>
    <operation>delete</operation>
    <node>nodeA.x.y.z</node>
    <interface>fa/09</interface>
    <node>nodeB.x.y.z</node>
    <interface>fa/05</interface>
  </connection>
</connectionedits>
```

Finally, run the command passing the `mychg.xml` with the `-f` option.

```
nnmconnect.ovpl -u username -p password -f mychg.xml
```

## AUTHOR

`nnmconnect.ovpl` was developed by Hewlett-Packard Company.

## FILES

None

## See Also

None

[Return to Reference Pages Index](#)

## Name

`nnmdeleteattributes.ovpl` — script to delete custom attributes from a comma separated values (CSV) file for both nodes and interfaces.

## SYNOPSIS

```
nnmdeleteattributes.ovpl [-?] [-u <username> -p <password>] [-t <object type>] [-f  
<csv_filename>] [-s <csv formatted line>]
```

## DESCRIPTION

`nnmdeleteattributes.ovpl` allows custom attributes to be deleted from a comma separated values (CSV) file, such as a .csv file from Microsoft™ Excel. This command is useful if you have previously created custom attributes that are no longer needed. This command will delete attributes from either nodes or interfaces. For nodes, once the attributes are deleted, any nodes in a node group formed by referencing those attributes will disappear from the node group.

You must either be logged on to the system as root/administrator, or supply a user/password with NNMi "Administrator" authority to run this command.

## Parameters

`nnmdeleteattributes.ovpl` supports the following options:

- ?  
Prints the usage statement.
- u <username>  
Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.
- p <password>  
Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.
- t <object type>  
Supply the object type to load attributes on. Must be either "node" or "interface".
- f <csv\_filename>  
Supply the CSV file name (with path, e.g. /opt/tmp/mynodes.csv) that lists the custom attributes to delete.
- s <csv formatted line>

Supply a single CSV formatted line. Eliminates the need to create a file for a simple change

## Syntax of Comma Separated File for Nodes

The CSV file you supply must have the following syntax for deleting attributes from nodes.

Empty lines are ignored.

Lines that begin with the # character are ignored.

- Column 1(A) : Node DNS|IP Address

Specify the DNS name of the node OR the IP address. This field is compulsory.

- Column 2(B) : Attribute Name

The name of the custom attribute.

Additional attribute names may be specified on the same line or on a separate line with the same node DNS|IP Address.

Example lines:

192.168.1.1,Project,Service Type

192.168.1.1,Asset Tracking

192.168.2.2,Project,Service Type,Asset Tracking

## Syntax of Comma Separated File for Interfaces

The CSV file you supply must have the following syntax for deleting attributes from interfaces.

Empty lines are ignored.

Lines that begin with the # character are ignored.

- Column 1(A) : Node DNS|IP Address

Specify the DNS name of the node OR the IP address. This field is compulsory.

- Column 2(B) : Interface Id

Specify the identifier of the interface on the node specified in the previous field. The value may be the interface index, alias, name or description and is searched in this same order. All matching interfaces have the attribute(s) loaded. This field is compulsory.

- Column 3(C) : Attribute Name

The name of the custom attribute.

Additional attribute names may be specified on the same line or on a separate line with the same node

Example lines:

```
192.168.1.1,1001,Project,Service Type
```

```
192.168.1.1,1001,Asset Tracking
```

```
192.168.2.2,A1,Project,Service Type,Asset Tracking
```

## Use of Microsoft Excel

Microsoft Excel is a handy tool to create comma separated files, but .csv files do not maintain the Excel spreadsheet's column width, comments, etc. It is recommended that you store a `nmdeleteattributes.ovpl` input file as a native .xls format, and then perform `File:Save As...` to create a .csv file. Then you can add Excel comments to the file, make columns wider, and you do not need to worry about escaping the comma character.

## EXAMPLES

Sample CSV file contents for nodes:

```
192.168.2.2,Project,Service Type,Asset Tracking
```

To delete the Node custom attributes from a CSV file:

```
nmdeleteattributes.ovpl -t node -u system -p myadminpasswd -f /tmp/test.csv
```

To delete the Node custom attributes from the command line:

```
nmdeleteattributes.ovpl -t node -u system -p myadminpasswd -s "192.168.1.1,Project"
```

Sample CSV file contents for interfaces:

```
192.168.2.2,1001,Project,Service Type,Asset Tracking
```

To delete the Interface custom attributes from a CSV file:

```
nmdeleteattributes.ovpl -t node -u system -p myadminpasswd -f /tmp/test.csv
```

To delete the Interface custom attributes from the command line:

```
nmdeleteattributes.ovpl -t node -u system -p myadminpasswd -s "192.168.1.1,Project"
```

## AUTHOR

`nmdeleteattributes.ovpl` was developed by Hewlett-Packard Company.

## FILES

```
$NNM_BIN/nmdeleteattributes.ovpl
```

## SEE ALSO

[nnmloadattributes.ovpl\(1M\)](#), [nnmloadnodegroups.ovpl\(1M\)](#), [nnm.properties\(4\)](#).

[Return to Reference Pages Index](#)

## Name

`nnmdeleteurlaction.ovpl` — delete URL actions and Menus for the supplied author key.

## SYNOPSIS

```
nnmdeleteurlaction.ovpl [-h | -help | -?] <authorKey>
```

## DESCRIPTION

`nnmdeleteurlaction.ovpl` is a Perl script that enables you to delete URL actions and Menus by providing the author key to actions and/or menus for.

NOTE: When performing an action/menu delete, you must restart the UI (sign out/sign in) for the changes to take effect.

This script is primarily intended to be used by NNMi iSPIs or third party integrations to remove (during uninstall) any actions or menus that the integration created.

## Parameters

`nnmdeleteurlaction.ovpl` supports the following options:

```
-? | -h | -help
```

Prints the usage statement.

## EXAMPLES

```
nnmdeleteurlaction.ovpl com.hp.nms.author.nas
```

Deletes the URL actions and Menus that are created when the NAS integration is installed.

## AUTHOR

`nnmdeleteurlaction.ovpl` was developed by Hewlett-Packard Company.

## SEE ALSO

[nnmconfigimport.ovpl\(1M\)](#), [nnmconfigexport.ovpl\(1M\)](#).

[Return to Reference Pages Index](#)

## Name

`nnmdisableperfspi.ovpl` — disables the optional HP iSPI for Performance Smart Plug In for use on the local NNM system

## SYNOPSIS

`nnmdisableperfspi.ovpl`

## DESCRIPTION

**In order for NNM to stop collecting performance data and passing it to the Performance SPI for aggregation and reporting, several configuration changes must take place.**

1. The SPI licenses must be uninstalled/disabled.
2. URL launch actions must be removed and will no longer show up in the NNM UI.
3. The Single Sign On service must be un-deployed from the NNM jboss server.

Disabling the license will prevent future data collection of performance metrics as well as disabling the ability to enable or disable performance monitoring of interfaces through monitoring configuration.

URL launch actions are removed from the NNM UI menu system. All NNM consoles that are currently running will need to exit and log back in before the action menu will reflect these changes. The URL actions that are removed are as follows: Reporting - Home Page, Reporting - Report Menu, Reporting - Data Explorer Live, Reporting - Chart Detail Live, Reporting - Path Health.

The `ssoservlet` that provides for single sign on between NNM and iSPI for Performance will be removed.

This script currently will not remove the user or shared drive that was setup during the `nnmenableperfspi.ovpl` script. These will need to be removed manually. Refer to the NNM iSPI for Performance documentation for additional information.

## ENVIRONMENT

This script must be run with full administrative privileges and requires standard NNM environment variables to be configured.

## AUTHOR

`nnmdisableperfspi.ovpl` was developed by Hewlett-Packard Company.

## FILES

The `nmdisableperfspi.ovpl` script removes the `ssoservlet.war` from the jboss structure.

## SEE ALSO

`nmlicense.ovpl(1M)` , `nmenableperfspi.ovpl(1M)` .

[Return to Reference Pages Index](#)

## Name

nnmdiscocfg.ovpl

## SYNOPSIS

```
nnmdiscocfg.ovpl -autodisco rule=ruleName rangetype=ignore/include [ -f ipAddressRangeFile | -n ipAddressRanges ] [-u <username> -p <password>] [-jndiHost <hostName> Default: localhost] [-jndiPort <port> Default: 1099]
```

```
nnmdiscocfg.ovpl -excludeipaddrs [ -f ipAddressRangeFile | -n ipAddressRanges ] [-u <username> -p <password>] [-jndiHost <hostName> Default: localhost] [-jndiPort <port> Default: 1099]
```

## DESCRIPTION

nnmdiscocfg.ovpl allows the addition of ip address ranges to existing autodiscovery rules. The autodiscovery ip address ranges control how discovery finds devices in your network.

Additionally, discovery IP address range exclusion filters can be added to prevent the creation of ip addresses in the NNMi topology. Addresses matching the filters will not be associated with nodes or interfaces and will not show up in the IP Address inventory. IP address range filters do not control how auto-discovery locates and identifies devices in your network.

## Parameters

The nnmdiscocfg.ovpl command recognizes the following parameters and options:

`-autodisco rule=ruleName rangetype=ignore/include`

Add ip address ranges to an existing auto-discovery rule specified by *ruleName*. The ranges can be included in the rule by specifying *rangetype=include* or ignored.

`-excludeipaddrs`

Add ip address ranges to the discovery 'Excluded IP Addresses' configuration.

`-f ipAddressRangeFile`

Specify a text file to read the IP address ranges from.

`-n ipAddressRanges`

Specify IP address ranges to load directly from the command line, with each range separated by a space.

`-u <username>`

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

Supply the NNMi administrator password required to execute the command. Required unless a [nmm.properties\(4\)](#) file exists.

-jndiHost <serverName>

The server JNDI host; default is localhost.

-jndiPort <port>

The server JNDI port; default is 1099.

## EXAMPLES

Add a list of IP address ranges to an existing autodiscovery rule:

```
nmmdiscocfg.ovpl -autodisco rule=bld1floor2 rangetype=include -n 15.2.112.21-34  
15.2.112.36 15.1.*.1-98
```

Add an IP address range to the IP Address exclusion configuration:

```
nmmdiscocfg.ovpl -u username -p password -excludeipaddrs -n 198.2.*.117
```

Load IP address range from a file named privateNet1.txt located on your local files system in the tmp directory:

```
nmmdiscocfg.ovpl -autodisco rule=bld1floor2 rangetype=exclude -f /tmp/lab3devices.txt
```

Load IP address range filters from a file named ignoreAddresses.txt located on your local files system in the tmp directory:

```
nmmdiscocfg.ovpl -excludeipaddrs -f /tmp/ignoreAddresses.txt
```

## AUTHOR

nmmdiscocfg.ovpl was developed by Hewlett-Packard Company.

## FILES

*Windows:* `install_dir\bin\nmmdiscocfg.ovpl`

*UNIX:* `$(NNM_BIN)/nmmdiscocfg.ovpl`

[Return to Reference Pages Index](#)

## Name

`nnmenableperfspi.ovpl` — enable the optional HP iSPI for Performance Smart Plug In for use on the local NNM system

## SYNOPSIS

`nnmenableperfspi.ovpl`

## DESCRIPTION

**In order for NNM to begin collecting performance data and pass it to the Performance SPI for aggregation and reporting, several configuration changes must take place.**

1. The SPI license must be enabled.
2. URL launch actions must be loaded into the NNM UI.
3. Drive space may need to be shared with the Performance SPI, possibly requiring a new operating system user to be configured on the NNM system.
4. The Single Sign On service must be deployed into the NNM jboss.

Enabling the license:

The license is initially enabled using an 'Instant On' setting which allows the Performance SPI to function for a limited period of time. The script achieves this by running the `nnmlicense.ovpl` command with the options `'PerfSPI -installIO'`

Loading URL Launch Actions:

Five separate launch actions are added into the NNM UI menu system. They are as follows: Reporting - Home Page, Reporting - Report Menu, Reporting - Data Explorer Live, Reporting - Chart Detail Live, Reporting - Path Health. The Path Health action is only available from the path view sections of the UI.

Sharing Drive Space (Required only when NNM and iSPI for Performance are on separate systems):

The directory where performance data is deposited by NNM is `$NnmInstallDir/data/shared/perfSpi/datafiles`. This must be shared with the Performance SPI system. When the Performance SPI and NNM co-exist on the same system there is no need to configure a network share of the directory structure. If NNM is installed on HP-UX, Solaris, or Linux the script will attempt to configure either NFS or Samba for the correct network share. NFS should be used if NNM and the iSPI for Performance are both on Unix (HP-UX, Solaris or Linux) systems. Samba should be used if NNM is on an HP-UX, Solaris, or Linux system and the iSPI for Performance is on a Windows system. If both NNM and the iSPI for Performance are on Windows the script can configure a Windows shared folder for you. When Windows file sharing or Samba is used to share the performance data folder, there is a requirement to have use a shared OS account (identical username/password) on both the NNM machine (for exporting the share) and on the iSPI for Performance machine (for accessing the share). The `nnmenableperfspi.ovpl` script will establish this user account for you on the NNM machine. Alternatively, the script offers you the opportunity to configure the operating

## Deploying the Single Sign On Service:

Users of NNM operate in an authenticated environment - they must provide a username and password in order to access the system. Those same credentials should be used to distinguish users of the Performance SPI system and to avoid double log in issues. In order for this to happen, NNM is configured as a 'trusted provider' for the Performance SPI. The SPI and NNM must be installed on systems which reside within the same network domain. The `ssoservlet` is part of the enabling apparatus for this functionality.

The following information will be required in order to complete the enablement:

- Fully qualified host name of the NNM system (Note: the enablement script will provide a default value which should work in most cases)
- Fully qualified host name of the SPI system - required only if the SPI will be installed on a remote server.
- Username and Password for enabling Windows or Samba file sharing - required only if the SPI will be installed on a remote \*Windows\* server.

## ENVIRONMENT

This script must be run with full administrative privileges and required standard NNM environment variables to be configured.

## AUTHOR

`nnmenableperfspi.ovpl` was developed by Hewlett-Packard Company.

## FILES

The `nnmenableperfspi.ovpl` script loads URL actions into the NNM menu system. These actions are stored as XML files in the `newconfig` directory of the NNM installation. It also places `ssoservlet.war` into the `jboss` structure. The war file is stored within the `newconfig` directory prior to deployment.

## SEE ALSO

`nnmlicense.ovpl(1M)` , `nnmconfigimport.ovpl(1M)` , `nnmconfigexport.ovpl(1M)` ,  
`nnmdisableperfspi.ovpl(1M)` .

[Return to Reference Pages Index](#)

## Name

`nnmhealth.ovpl` — Prints NNMi health information

## SYNOPSIS

```
nnmhealth.ovpl [-u <username> -p <password>] ( -print [quiet|brief|detailed|conclusions|verbose] | -activate <conclusions> | -suppress <conclusions> )
```

## DESCRIPTION

`nnmhealth.ovpl` prints out information about the internal health of NNMi. It support several levels of output from no output except for the return code (quiet) to the full health report (verbose).

In addition to viewing the current health of NNMi it may also be used to either suppress or re-activate individual health conclusions. Health conclusions may be suppressed by the administrator if they are aware of the problem and don't wish warnings to be shown until the issue is resolved.

Note that modifying the suppressed list using this tool will take effect on the next health scan and they will continue to be suppressed until the admin re-activates them or until NNMi is restarted.

## Parameters

`nnmhealth.ovpl` supports the following options:

`-help`

Prints the usage statement.

`-u <username>`

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-p <password>`

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-print <level>`

Prints information about the health of NNMi. The level can be one of the following:  
quiet|brief|detailed|conclusions|verbose

`-suppress <conclusions>`

Configures the supplied conclusions to be suppressed until NNMi is next restarted or until the conclusion is re-activated. The conclusions are specified as a comma separated list and active conclusions may be found using the `-print conclusions` option.

Note that a conclusion being suppressed does not imply it would otherwise be active.

`-activate <conclusions>`

Removes the supplied conclusions from the suppressed list. The conclusions are specified as a comma separated list and suppressed conclusions available to re-activate may be obtained using the `-print conclusions` option.

## EXAMPLES

```
nnmhealth.ovpl -u username -p password -print brief
```

Prints the overall status of NNMi.

```
nnmhealth.ovpl -u username -p password -print detailed
```

Prints the current list of health warnings.

```
nnmhealth.ovpl -u username -p password -suppress "SystemLowSwap, SystemLowSwapPercent"
```

Configures NNMi to skip checking the health of swap space in absolute terms and also as a percentage.

```
nnmhealth.ovpl -u username -p password -activate "SystemLowSwap"
```

Configures NNMi resume checking for the SystemLowSwap conclusion.

## AUTHOR

`nnmhealth.ovpl` was developed by Hewlett-Packard Company.

## SEE ALSO

[Return to Reference Pages Index](#)

## Name

nnmincidentcfg.ovpl — create incident configurations from SNMP MIBs

## SYNOPSIS

```
nnmincidentcfg.ovpl [ [ [-loadTraps mib_file [-authorLabel author_label -authorKey author_key]] | -
loadMib mib_file | -deleteAuthor author_key | -deleteCategory category_key | -deleteFamily
family_key | -disableAllTraps <true/false> [-u username] [-p password] ]
```

## DESCRIPTION

nnmincidentcfg.ovpl is used to create incident configurations for SNMP traps defined in a TRAP-TYPE or NOTIFICATION-TYPE macro in an SNMP MIB file. To load a MIB into NNMi for defining MIB Expressions or to display numeric SNMP Object Identifiers as text, use the nnmloadmib.ovpl command.

The created incident will have the following values, which can then be manually updated with the NNMi user interface:

1. Name will be set to the name of the trap/notification in the MIB file.
2. Oid will be set to the oid of the trap/notification in the MIB file.
3. Enable will be set to "true."
4. Category will be set to "Status."
5. Family will be set to "Node."
6. Severity will be set to "Normal."
7. Message Format will be set to the name of the incident configuration.
8. Description will be set to the trap/notification description in the MIB file.

The created incident can be accessed using the Incident Configuration view. This can be further customized as required.

nnmincidentcfg.ovpl supports a special annotation called *#SUMMARY*. The value for the *#SUMMARY* annotation will be used as Message Format value in the created incident configuration entry. This annotation is applied to the MIB file just after the trap description as a MIB comment. The following shows an example of this:

```
MyTrap TRAP-TYPE
ENTERPRISE hp
VARIABLES {
    serverName, trapTime, volumeName, volumeNum
}
DESCRIPTION "The disk volume is out of space. Please consult
your sysop, and/or the proper manual."

--#SUMMARY "Volume $1 on system $2 is out of space."
```

## Parameters

`nnmincidentcfg.ovpl` supports the following options:

`-loadTraps <mib_file>`

Specify the MIB file that has the trap definitions. `nnmincidentcfg.ovpl` parses trap/notification definitions (`TRAP-TYPE` or `NOTIFICATION-TYPE` macros) found in the MIB file, and creates incident configurations for each entry.

`-authorLabel <author_label>`

Specifies the label of the author for the target incident configurations. This is an optional parameter. If author label is specified, then author key must also be specified.

`-authorKey <author_key>`

Specifies the key of the author for the target incident configurations. This is an optional parameter. If author key is specified, then author label must also be specified. It is recommended that java packaging notation with your company's domain be used, such as `com.example.nnm.author`.

`-loadMib <dependent_mib_file>`

In some cases, the `mib_file` specified with the `-loadTraps` option may have dependencies on other MIBs that it imports. In such a case, the dependency has to be explicitly loaded using this option.

`-deleteAuthor <author_key>`

It may be desirable to delete an Author that is no longer being used by the incident configuration. This option may be used to delete an Author by specifying the Author key value as long as no configurations reference the Author object.

`-deleteCategory <category_key>`

It may be desirable to delete a Category that is no longer being used by the incident configuration. This option may be used to delete a Category by specifying the Category key value as long as no incidents or incident configurations reference the Category object.

`-deleteFamily <family_key>`

It may be desirable to delete a Family that is no longer being used by the incident configuration. This option may be used to delete a Family by specifying the Family key value as long as no incidents or incident configurations reference the Family object

`-disableAllTraps <true/false>`

If `true` all traps should be loaded as disabled in the incident configuration. The default value is `false`, meaning the incident configuration is enabled by default.

`-u <username>`

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-p <password>`

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

## EXAMPLES

```
nnmincidentcfg.ovpl -loadTraps CISCO-VTP-MIB.my -authorLabel "Cisco" -authorKey  
com.example.cisco.nnm.author
```

```
nnmincidentcfg.ovpl -loadTraps "C:\Other Mibs\mib-mpls" (Windows)
```

```
nnmincidentcfg.ovpl -loadTraps "/tmp/OtherMibs/mib-mpls" (UNIX)
```

## AUTHOR

nnmincidentcfg.ovpl was developed by Hewlett-Packard Company.

## SEE ALSO

*RFC 2578 Structure of Management Information Version 2 (SMIv2)*

*RFCs 1155, 1212, 1215: SNMP Version 1 Structure of Management Information*

*RFCs 1902, 1903, 1904: SNMP Version 2 Structure of Management Information*

[nnmloadmib.ovpl\(1M\)](#),

[nnm.properties\(4\)](#).

[Return to Reference Pages Index](#)

## Name

nnmldap.ovpl — reload or view LDAP configuration.

## SYNOPSIS

```
nnmldap.ovpl -reload | -info | -diagnose <username>
```

## DESCRIPTION

nnmldap.ovpl is a Perl script that enables you to reload, view or diagnose changes to the Lightweight Directory Access Protocol (LDAP) sign-in configuration without restarting jboss.

## Parameters

nnmldap.ovpl supports the following options:

-info

Displays the LDAP configuration, such as

```
Configuration=providerURL:"ldap://example.com:636/". Number of available Incident  
assignment users:0
```

-reload

Reloads the LDAP configuration.

-diagnose <username>

Verifies configuration in the ldap.properties file by attempting to access <username> in the Directory Service using the LDAP configuration parameters. This command will respond with information to help you diagnose LDAP configuration problems.

<username> must be a valid username in the Directory Service. It is the same name that is used in the NNMi console username prompt of the NNMi login screen.

## EXAMPLES

```
nnmldap.ovpl -info
```

Returns the current LDAP configuration.

```
nnmldap.ovpl -reload
```

Reads modifications to the ldap.properties file (such as enabling or disabling LDAP).

```
nnmldap.ovpl -diagnose <username>
```

Shows configuration parameters for the ldap.properties file, and verifies that information can be extracted from the Directory Service.

## AUTHOR

nnmlldap.ovpl was developed by Hewlett-Packard Company.

## SEE ALSO

[ldap.properties\(4\)](#).

[Return to Reference Pages Index](#)

## Name

nnmlicense.ovpl — administer Network Node Manager licensing

## SYNOPSIS

```
nnmlicense.ovpl [-h | -help]
```

```
nnmlicense.ovpl [ <PRODUCT> [-nosync] [(-g|-gui)|(-install|-f <lic_file>)] ]
```

The `-f` | `-install` option installs license passwords into the licensing database.

## DESCRIPTION

`nnmlicense.ovpl` provides license management for HP Network Node Manager (NNMi). License management includes the ability to retrieve license passwords from Hewlett-Packard, installing license passwords from a file, the removal of license passwords, and reporting valid licenses.

There are two steps to adding a license: updating the license database and notifying the running NNMi processes that new license information is available. The `-nosync` option performs the former but does not notify NNMi. If the `-nosync` option is not specified, the running NNMi system is automatically notified; it is not necessary to stop and restart NNMi.

The `-nosync` option allows you to perform multiple licensing tasks (e.g. removing a license and installing a replacement license). You then notify NNMi with the following:

```
nnmlicense.ovpl NNM
```

The `-g` | `-gui` option opens the license management GUI, which provides access to most license management capabilities:

### Retrieve/Install License Key

Enables a license password to be retrieved from Hewlett-Packard.

### Install/Restore License Key

Installs license passwords from a text file.

### Request License Key through Email/Fax

Provides the ability to request a license for systems that cannot transfer data to Hewlett-Packard.

### Report License Key

Reports license information for the particular product.

### Backup License File

Enables the backing up of a license file before installing or removing licenses.

Removes license keys (usually temporary or emergency licenses provided by Hewlett-Packard).

### Recover License Key

Adds back licenses that were previously removed.

The license management GUI enables multiple operations before updating the running NNMi system. For example, you can replace a license by removing the old license, and then adding a new one in its place.

## Parameters

### *PRODUCT*

The short name for the product being licensed.

### *-nosync*

Prevents synchronization with the running NNMi system.

### *-g|-gui*

Starts the license management GUI.

### *-f|-install <lic\_file>*

Install the license contained in the specified license file

## EXAMPLES

To install a license password contained in a file named "license.txt", perform the following:

```
$NnmInstallDir/bin/nmlicense.ovpl NNM -f license.txt
```

This will update the license database and notify NNMi of the licensing change. To avoid notifying NNMi, the *-nosync* option can be provided:

```
$NnmInstallDir/bin/nmlicense.ovpl NNM -nosync -f license.txt
```

The following example demonstrates replacing a license with a new one. Suppose the customer is given a temporary license for the NNM product, with a capacity of 500 nodes. Sometime later this customer receives their official license. The temporary license must first be removed the permanent license is accepted. This can be done using the license management GUI. First remove the temporary license (Remove License Key), and then add the permanent license (Install/Restore License Key). The running NNMi system is updated when they exit the GUI.

## AUTHOR

`nmlicense.ovpl` was developed by Hewlett-Packard Company.

## FILES

`$NnmInstallDir/misc/nms/lic/NNM.pdf`

Product definition file used by licensing.

`$NnmDataDir/shared/nm/conf/licensing/NNM.bin`

Data file representing the license information that is consumed by `ovjboss`.

`<drive>:\Program Files (x86)\Common Files\Hewlett-Packard\HPOvLIC`

Program files and data files for licensing on Microsoft Windows systems.

`/opt/OV/HPOvLIC`

Program files for licensing on UNIX-based systems.

`/var/opt/OV/HPOvLIC`

Data files for licensing on UNIX-based systems.

## SEE ALSO

Installation Guide for future details on licensing.

[Return to Reference Pages Index](#)

## Name

`nnmloadattributes.ovpl` — script to load custom attributes from a comma separated values (CSV) file for nodes and interfaces.

## SYNOPSIS

```
nnmloadattributes.ovpl [-?] [-u <username> -p <password>] [-t <object type>] [-f  
<csv_filename>] [-r <true / false>] [-s <csv formatted line>]
```

## DESCRIPTION

`nnmloadattributes.ovpl` allows custom attributes to be loaded from a comma separated values (CSV) file, such as a `.csv` file from Microsoft™ Excel. This command is useful if you have a large number of nodes or interfaces that are defined in an external datastore, and you would like to load these attributes into NNM. For nodes, after loading into NNM you can use the Node Group forms to group nodes according to their custom attributes.

You must be logged on as root/administrator to run this command.

## Parameters

`nnmloadattributes.ovpl` supports the following options:

`-?`

Prints the usage statement.

`-u <username>`

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-p <password>`

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-r <true / false>`

Existing attribute values are not changed unless this argument is provided with the value of true.

`-t <object type>`

Supply the object type to load attributes on. Must be either "node" or "interface".

`-f <csv_filename>`

Supply the CSV file name (with path, e.g. `/opt/tmp/mynodes.csv`) from where you want to import the custom attributes.

`-s <csv formatted line>`

Supply a single CSV formatted line. Eliminates the need to create a file for a simple change

## Syntax of Comma Separated File For Node Attributes

The CSV file you supply must have the following syntax for adding attributes on nodes. See below for the syntax for interface attributes.

Empty lines are ignored.

Lines that begin with the # character are ignored.

- Column 1(A) : Node DNS|IP Address

Specify the DNS name of the node OR the IP address. This field is compulsory.

- Column 2(B) : Attribute Name

The name of the custom attribute.

- Column 3(C) : Attribute Value

The value of the custom attribute.

Additional attribute name and value pairs may be specified on the same line or on a separate line with the same node DNS|IP Address.

Example lines:

192.168.1.1,Location,Building Five Upper,Service Type,eCommerce

192.168.1.1,Asset Tracking,N1234

192.168.2.2,Location,Fort Collins,Service Type,IT,Asset Tracking,F4321

## Syntax of Comma Separated File For Interface Attributes

The CSV file you supply must have the following syntax for adding attributes on interfaces..

Empty lines are ignored.

Lines that begin with the # character are ignored.

- Column 1(A) : Node DNS|IP Address

Specify the DNS name of the node OR the IP address. This field is compulsory.

- Column 2(B) : Interface Id

Specify the identifier of the interface on the node specified in the previous field. The

value may be the interface index, alias, name or description and is searched in this same order. All matching interfaces have the attribute(s) loaded. This field is compulsory.

- Column 3(C) : Attribute Name

The name of the custom attribute.

- Column 4(D) : Attribute Value

The value of the custom attribute.

Additional attribute name and value pairs may be specified on the same line or on a separate line with the same node DNS|IP Address and Interface Id.

Example lines:

```
192.168.1.1,1001,Location,Building Five Upper,Service Type,eCommerce
```

```
192.168.1.1,1001,Asset Tracking,N1234
```

```
192.168.2.2,A1,Location,Fort Collins,Service Type,IT,Asset Tracking,F4321
```

## Use of Microsoft Excel

Microsoft Excel is a handy tool to create comma separated files, but .csv files do not maintain the Excel spreadsheet's column width, comments, etc. It is recommended that you store a `nnmloadattributes.ovpl` input file as a native .xls format, and then perform `File:Save As...` to create a .csv file. Then you can add Excel comments to the file, make columns wider, and you do not need to worry about escaping the comma character.

## EXAMPLES

Sample node CSV file contents:

```
192.168.2.2,Location,Fort Collins,Service Type,IT,Asset Tracking,F4321
```

To load the Node custom attributes from a CSV file overwriting existing values:

```
nnmloadattributes.ovpl -t node -u system -p myadminpasswd -f /tmp/test.csv -r true
```

To load the Node custom attributes from the command line:

```
nnmloadattributes.ovpl -t node -u system -p myadminpasswd -s "192.168.1.1,Project,IT  
Update of Building Five"
```

Sample interface CSV file contents:

```
192.168.2.2,A1,Location,Fort Collins,Service Type,IT,Asset Tracking,F4321
```

To load the Interface custom attributes from a CSV file overwriting existing values:

```
nnmloadattributes.ovpl -t interface -u system -p myadminpasswd -f /tmp/test.csv -r true
```

To load the Interface custom attributes from the command line:

```
nmloadattributes.ovpl -t interface -u system -p myadminpasswd -s  
"192.168.1.1,1001,Project,IT Update of Building Five"
```

## AUTHOR

nmloadattributes.ovpl was developed by Hewlett-Packard Company.

## FILES

`$NNM_BIN/nmloadattributes.ovpl`

## SEE ALSO

[nmdeleteattributes.ovpl\(1M\)](#), [nmloadnodegroups.ovpl\(1M\)](#), [nnm.properties\(4\)](#).

[Return to Reference Pages Index](#)

## Name

nnmloadmib.ovpl — load and unload SNMP MIBs

## SYNOPSIS

```
nnmloadmib.ovpl [ [-load mib-file] [-unload mib-file] [-list] [-resynch] [-u userName] [-p password] ]
```

## DESCRIPTION

nnmloadmib.ovpl loads an SNMP MIB (Management Information Base) in the Internet SMI (Structure of Management Information) format for use by NNMi applications when converting SNMP Object Identifiers (OIDs) from numeric to human readable text. MIBs must also be loaded before a MIB Expression can be created using the NNMi user interface. NNMi support SMI Version 1 (RFC1155, 1212, 1215) and SMI Version 2 (RFC2578) SMI formats.

nnmloadmib.ovpl maintains the ASCII file `snmpmib`, which is a collection of MIB modules, and compiles it to produce `snmpmib.bin` for use by NNMi applications and stores the resulting information in the main database.

To load TRAP-TYPE and NOTIFICATION-TYPE macros from a MIB into the NNMi Incident Configuration, use the `nnmincidentcfg.ovpl` command.

## Parameters

nnmloadmib.ovpl supports the following parameters:

`-load mib-file`

Load *mib-file* into the MIB database.

Note: HP recommends first copying the *mib-file* to the `/var/opt/OV/shared/nnm/user-snmp-mibs` (UNIX) or `%NnmDataDir%\shared\nnm\user-snmp-mibs` (Windows) directory (or a child directory) before running the **nnmloadmib.ovpl** command. This enables the **Actions -> Display MIB File** menu and the `-list` option to find the location of the original MIB file.

`-unload mib-file`

Unload *mib-file* from the MIB database, where *mib-file* is the `basename(1)` of the *mib-file* originally loaded into the MIB database.

`-resynch`

Re-synchronizes the NNMi database with the compiled MIB file `snmpmib.bin`.

This command is only needed if you have replaced MIBs with vastly different contents, or if the MIB Variables table view appears to be out of sync with the MIB files that have been loaded.

Lists the MIBs loaded in the database.

This option does not require the user to provide a username and password.

`-u <username>`

Supply the NNMi administrator username required to execute the command. Required unless a [nmm.properties\(4\)](#) file exists.

`-p <password>`

Supply the NNMi administrator password required to execute the command. Required unless a [nmm.properties\(4\)](#) file exists.

## Syntax of MIB Files

Most of the relevant syntax for MIB files is described in various RFC documents. See the "SEE ALSO" section of this reference page (and the UNIX manpage).

## Diagnostics

`nmmloadmib.ovpl` returns the following exit codes:

0

Operation was successful.

1

Invalid command-line usage.

2

Syntax error was detected in the MIB file.

3

MIB file was already loaded, and `-replace` was not used (`nmmloadmib.ovpl` always passes `-replace` to the MIB compiler).

4

MIB name specified by `-unload` was not loaded.

5

MIB file cannot be accessed.

6

MIB file specified with `-load` is not a file.

7

Error accessing an MIB database.

8

Error creating an MIB database.

9

Error creating a compiled form MIB database.

10

Error creating a working temporary MIB database.

11

`nnmloadmib` was interrupted by a signal.

12

`nnmloadmib` is out of memory.

21

`nnmloadmib` was provided an incorrect username or password. Run again with the `-resynch` option.

22

`nnmloadmib` failed with an unexpected error (details in the message). Run again with the `-resynch` option.

## EXAMPLES

To load the MIB file `/var/opt/OV/shared/nnm/user-snmp-mibs/corp.mib`:

```
nnmloadmib.ovpl -load /var/opt/OV/shared/nnm/user-snmp-mibs/corp.mib -u user -p password
```

To list the loaded MIBs:

```
nnmloadmib.ovpl -list -u user -p password
```

## AUTHOR

`nnmloadmib.ovpl` was developed by Carnegie-Mellon University and Hewlett-Packard Company.

## FILES

`$NnmInstallDir/lbin/nnmloadmib`

`$NnmDataDir/shared/nnm/conf/snmpmib`

`$NnmDataDir/shared/nnm/conf/snmpmib.bin`

`$NnmDataDir/shared/nnm/conf/snmpmibmod_var`

`$NnmInstallDir/misc/nnm/snmp-mibs/*`

`$NnmDataDir/shared/nnm/user-snmp-mibs/*`

## SEE ALSO

*RFC 2578 Structure of Management Information Version 2 (SMIv2)*

*RFCs 1155, 1212, 1215: SNMP Version 1 Structure of Management Information*

*RFCs 1902, 1903, 1904: SNMP Version 2 Structure of Management Information*

[nnmincidentcfg.ovpl](#)(1M),

[nnmsnmpwalk.ovpl](#)(1M).

[Return to Reference Pages Index](#)

## Name

nnmloadnodegroups.ovpl — script to load Node Group definitions from a comma-separated values (CSV) file

## SYNOPSIS

```
nnmloadnodegroups.ovpl [-?] [-u <username> -p <password>] [-r true | false] -f <csv_filename>
```

## DESCRIPTION

nnmloadnodegroups.ovpl allows Node Group definitions to be loaded from a comma-separated values (CSV) file, such as a .csv file exported from Microsoft™ Excel. This command is useful if you have a large amount of Node data defined in an external datastore, and you would like to load it into the NNMi database as a starting point for Node Group definitions. After loading into NNMi, you can use the Node Group form to further refine the definition of each Node Group.

The following settings cannot be set in the CSV file. You must import the Node Group and then use the Node Group form to modify these default settings:

- Calculate Status = true (NNMi calculates the Node Group status)
- Add to Filter List = false (NNMi does not add this Node Group to NNM iSPI Performance for Metrics and NNM iSPI Performance for Traffic)

You must be logged on as root user or NNMi administrator to run this command.

## Parameters

nnmloadnodegroups.ovpl supports the following options:

-?

Prints the usage statement.

-u <username>

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

-p <password>

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

-r true | false

Back up the existing Node Group configuration before using this option.

-r false (the default setting) means if the Node Group Name already exists in the NNMi database, the nnmloadnodegroups.ovpl command does not change the previous settings.

-r true means all the settings for any existing Node Group with the same Name (column 1) are overwritten with the values in your CSV file. Caution: this is not a merge, it is a complete replacement of that Node Group configuration.

-f <csv\_filename>

Enter the CSV file name and the path for the CSV file.

## Syntax of Comma-Separated File

The CSV file you supply must have the following syntax:

Required. Column 1. You must provide a value for Node Group Name.

Optional. Columns 2-11 are optional. Leave any combination of these columns blank.

No comma is required to indicate the end of Column 11. No semi-colon ";" is required at the end of the last entry within a column (between commas ",").

NNMi combines the results of all settings in the following manner:

1. NNMi first evaluates any Device Filters (column 5). Nodes must match at least one specification to belong to this Node Group.

2. NNMi then evaluates any Additional Filters (columns 7-11). Nodes must also pass all Additional Filters specifications to belong to this Node Group. Note: After importing, the Node Group form's Additional Filters tab displays the combined columns 7-11 settings. You can change the default Boolean logic using the Node Group form's Additional Filters Editor.

3. Any Additional Nodes specified (column 6) are always included in the Node Group, regardless of any filters.

4. Any Child Node Groups (column 4) results are treated the same as Additional Nodes.

Empty lines or lines starting with a # character are ignored as comments. Add the following comment to line 1 to make it easy to remember the syntax for each required column.

```
#[NodeGroupName],[Notes],[AddtoFilterList],[ChildNodeGroup:0/1;...],DeviceFilter[Category1:Vendor1:Family1:Profile1;...],AdditionalNodes[Fully-Qaul-hostname;...],AdditionalFilters > [hostname;...],[hostedIPAddress;...],[mgmtIPAddress;...],[customAttrName/customAttrValue;...],[capability;...]
```

- Column 1(A) : Node Group Name

Required. Specify the Name of the Node Group you want to import. (This becomes the Name attribute value in the Node Group form.)

- Column 2(B) : Notes

Optional. Describe the Node Group in your own terms. (This becomes the Notes field text in the Node Group form.)

- Column 3(C) : Add to View Filter List

Optional. Sets the Add to View Filter List field of the Node Group form.

1 (the default setting) means this Node Group is available in the drop-down filter list when viewing tables, such as the All Nodes table.

0 means do not include this Node Group in the view drop-down filter list.

Recommendation: Set this value to 1 only for top-level or most-commonly-used Node Groups. Avoid too many Node Groups or the view filter list is too long and difficult to use.

- Column 4(D) : Child Node Groups

Optional. Specify a list of child Node Groups for this Node Group, separated by semi-colon ";" (After importing, these specifications appear on the Child Node Groups tab of the Node Group form.) Note: If you are configuring Child Node Groups, the specified child Node Group must either already exist in the NNMi database or be defined in the same CSV file.

Example: `ChildNodeGroup1:1[;ChildNodeGroup2:0;...]`

0 (the default setting) means that child Node Group is shown as a Node Group icon in maps of the parent Node Group.

1 means expand the child Node Group in a map of the parent Node Group. This option displays all nodes as if they were defined within the parent Node Group.

Valid entries for Child Node Groups are:

- computers:1
- computers:0
- computers:
- computers::printers:1

- Column 5(E) : Device Filters

Optional. Add Device Filters settings, separated by semi-colon ";" (After importing, these specifications appear on the Device Filters tab of the Node Group form.) Each filter specification consists of 4 optional colon-separated parts in the following format:

```
Category1:Vendor1:Family1:Profile1[;Category2:Vendor2:Family2:Profile2 ...]
```

Provide the exact specification from the device's MIB file (not the text string displayed in the NNMi console's Device Profile form).

To match more filters, you may omit portions of a filter specification. For example, if you want to match any family for Category1 and Vendor1, add an entry such as the following:

```
Category1:Vendor1::
```

To leave family unspecified for filter1 and family and profile unspecified for filter2:

```
Category1:vendor1::profile1;Category2:vendor2::;
```

Valid example entries for device profile:

- `com.hp.ov.nms.devices.printer:com.hp.ov.nms.devices.hewlettpackard::1.3.6.1.4.1.9.1.380`
- `com.mycomp.ov.nms.devices.printer:com.hp.ov.nms.devices.mycompanyname::`
- `com.hp.ov.nms.devices.printer::`
- `:::1.3.6.1.4.1.9.1.380`

- Column 6(F) : Additional Nodes

Optional. Specify a list of specific node hostnames that you want to add to this Node Group, separated by semi-colon ";" (After importing, these specifications appear on the Additional Nodes tab of the Node Group form.) The hostnames you provide must be the current value of the fully-qualified, case-sensitive Hostname attribute as it appears on the Node form.

For example `hostname1.x.y.z;hostname2.x.y.z;hostname3.x.y.z`

- Column 7(G) : Additional Filters "sysName" code (Hostname Wildcards)

Optional. List the hostname wildcards separated by semi-colon ";" (equivalent to the Operator "="). If you need other Operators, use the Node Group form after importing (these specifications appear on the Additional Filters tab of the Node Group form).

For example: `*.cnd.hp.com;*snmp.hp.com`

- Column 8(H) : Additional Filters "hostedIPAddress" code (Hosted IP Address Ranges)

Optional. List the hosted IP address ranges separated by semi-colon ";" (equivalent to the Operator "=") If you need other Operators, use the Node Group form after importing (these specifications appear on the Additional Filters tab of the Node Group form). Ranges have a lower and an upper address, separated by a dash. The addresses are inclusive. To include a single IP address, use the same value for the lower and upper address values. Note that if any address on a node matches this range, the node will be included in the Node Group.

Valid example: `10.20.30.1-10.20.30.254;192.168.177.1-192.168.180.254;1.1.1.1-1.1.1.1`

- Column 9(I) : Additional Filters "mgmtIPAddress" code (Management Address Ranges)

Optional. List the management Address ranges separated by semi-colon ";" (equivalent to the Operator "=") Ranges are in the same format as hosted IP address ranges. If you need other Operators, use the Node Group form after importing (these specifications appear on the Additional Filters tab of the Node Group form). Note that Spiral Discovery only creates management IP Addresses on nodes that support SNMP. See the online help for the Node form's Management Address field for more information about how Spiral Discovery selects the Management Address.

- Column 10(J) : Additional Filters "customAttrName:customAttrValue" codes (Custom Node Attributes)

Optional. List the custom attributes assigned to nodes as follows: "custom attribute name" operator "custom attribute value"[;... ] and note the name and the value must be surrounded by quotes. After importing, these Custom Node Attribute specifications appear on the Additional Filters tab of the Node Group form.

Valid values for operator are as follows:

`=, !=, like, not like, between, not between, >, >=, <, <=, is null, is not null` (If you need other values, use the Node Group form after importing.) The operators `is null` and `is not null` do not have a value, for example, "my attribute" is not null. The values for `between` and `not between` are specified as `x AND y` (for example, "my attribute" between "100 AND 200").

For more than one custom attribute statement, place a semi-colon between statements (for example, "Location" = "Bldg. Five"; "Service Type" = "eCommerce"). Multiple "customAttrName:customAttrValue" statements are AND'ed together. Therefore, all the statements must evaluate to true for each node to be included in the Node Group.

- Column 11(K) : Additional Filters "capability" code (Capabilities)

Optional. List the capabilities assigned to nodes as follows: `capability operator "capability value"[;... ]` Note the value must be surrounded by quotes. After importing, these Capabilities specifications appear on the Additional Filters tab of the Node Group form.

The valid values for operator are as follows:

`=, !=, like, not like` (If you need other values, use the Node Group form after importing.)

For more than one capability statement, place a semi-colon between statements (for example, `capability = "com.hp.ov.nms.isLANSwitch";capability != "com.hp.ov.nms.isIPv4Router"`). Multiple capability statements are AND'ed together. Therefore, all the statements must evaluate to true for each node to be included in the Node Group.

## Use of Microsoft Excel

Microsoft Excel is a useful tool to create comma-separated files, but .csv files do not maintain their column width, comments, etc. It is recommended that you store a nnmloadnodegroups.ovpl input file as a native .xls format, then perform `File:Save As...` to create a .csv file. You can then add Excel comments to the file, make columns wider, and not need to worry about escaping the comma character. Microsoft Excel also makes it easy to populate the list of child Node Groups. Simply make column 4(D) contain a calculated value such as the following:

```
= $A1&" : 0 ; "&$A2&" : 0 ; "&$A3&" : 0 ; "&$A4&" : 0 ; "&$A5&" : 0 ; "&$A6&" : 0 ; "&$A7&" : 0 ; "
```

This example combines the Node Group Names defined in the first column of the first 7 rows, as child Node Groups of the Node Group Name defined in the first cell of the current row. Using this Excel reference, if you rename the child Node Group in the first column, you do not need to go back and change the reference in the parent Node Group's column 4(D). Note that typing a comma in a line after a leading # in Microsoft Excel generates a non-commented entry when the .xls file is saved as a .csv file (creating a Node Group with a Name starting with the # character).

## EXAMPLES

Sample CSV file contents:

```
SNMP,Nodes that support SNMP and that are present in Colorado,,,,,server1.myco.com;server2.myco.com,* .hp.com
```

### Note

When entering data in CSV files, do not use the separator characters (":" and ";") for other purposes (for example, in the Names - column 1 - of Node Groups). If you need to use the separator characters, escape them with "\". For example:

- "computer:1" must be entered as "computer\;1"
- "computer;1" must be entered as "computer\;"
- "computer\;1" must be entered as "computer\\;1"

To load the Node Groups from a CSV file without overwriting any existing Node Group that matches a Name defined in column 1 of your CSV file:

```
nnmloadnodegroups.ovpl -u system -p myadminpasswd -f /tmp/test.csv
```

To load the Node Groups from a CSV file, and overwrite any existing Node Group that matches a Name defined in column 1 of your CSV file (Caution: this is not a merge, it is a complete replacement of that matching Node Group's configuration):

```
nnmloadnodegroups.ovpl -u system -p myadminpasswd -r true -f /tmp/test.csv
```

## AUTHOR

nnmloadnodegroups.ovpl was developed by Hewlett-Packard Company.

## FILES

```
$NNM_BIN/nnmloadnodegroups.ovpl
```

## SEE ALSO

[nnmconfigimport.ovpl\(1M\)](#), [nnm.properties\(4\)](#).

[Return to Reference Pages Index](#)

## Name

nnmloadseeds.ovpl — load discovery node seed information

## SYNOPSIS

```
nnmloadseeds.ovpl -f seedFile | -n seeds [-u <username> -p <password>] [-jndiHost <hostName>  
Default: localhost] [-jndiPort <port> Default: 1099]
```

## DESCRIPTION

nnmloadseeds.ovpl allows discovery seeds to be loaded using command line arguments (-n option) or from a text file (-f option). A seed is a device that you want NNM to use as a starting point for the spiral discovery process. Seed values are either IP addresses or host names. When using the -n option the seeds are entered on the command line, separated by white space. Seeds are always added to NNM even if they do not support SNMP

The -f option accepts a file with a single entry specified per line. Each line has the following format:

```
IPAddress/HostName # (optional comment to help identify the node, if desired)
```

Where:

*IPAddress* = the IP address of the node you wish to add.

*HostName* = the host name of the node you wish to add.

Comments can be delimited with a # character. Additionally, you can use INCLUDE-FILE *filename* to include other seedfiles.

Note that you should set up the SNMP configuration for the devices being loaded before running this command.

## Parameters

The nnmloadseeds.ovpl command recognizes the following parameters and options:

-f *seedFile*

Specify a text file to read the seeds from.

-n *seeds*

Specify seeds to load directly from the command line, with each seed separated by a space.

-u <username>

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-p <password>`

Supply the NNMi administrator password required to execute the command. Required unless a [nmm.properties\(4\)](#) file exists.

`-jndiHost <serverName>`

The server JNDI host; default is localhost.

`-jndiPort <port>`

The server JNDI port; default is 1099.

## EXAMPLES

Load a list of devices as seeds:

```
nnmloadseeds.ovpl -n mimcisco8540 15.2.112.22
```

Load seed from a node using its fully qualified name with username and password:

```
nnmloadseeds.ovpl -u username -p password -n mimcisco8540.superpoller3.mim
```

Load seeds from a file name that is named seeds\_to\_load.txt:

```
nnmloadseeds.ovpl -f /tmp/seeds_to_load.txt
```

## AUTHOR

nnmloadseeds.ovpl was developed by Hewlett-Packard Company.

## FILES

*Windows:* `install_dir\bin\nnmloadseeds.ovpl`

*UNIX:* `$NNM_BIN/nnmloadseeds.ovpl`

[Return to Reference Pages Index](#)

## Name

`nnmmanagementmode.ovpl` — Change the NNM management mode of a node or interface.

## SYNOPSIS

```
nnmmanagementmode.ovpl [[-node.name nodename -mode mode]
| [-t object type [[ -f csv_filename [-b batch size ] ] | [-s csv formatted line ] ] ]
[-jdnHost hostname]
[-jdnPort port]
[-u username -p password]
[-?]
```

## DESCRIPTION

`nnmmanagementmode.ovpl` allows the system administrator to set the management mode of a node or interface in the NNM database.

## Parameters

`nnmmanagementmode.ovpl` recognizes the following options. Any unrecognized options are reported by a usage message.

-?

Prints the usage statement.

-node.name *nodename*

The name of a node to change management mode on.

-mode *mode*

The desired management mode to set. Valid values are "MANAGED", "NOTMANAGED", or "OUTOFSERVICE".

-t *object type*

Supply the object type to set the management mode on. Valid values are "node" or "interface".

-f *csv\_filename*

Supply the CSV file name (with path, e.g. /opt/tmp/mynodes.csv) from where you want to set the management mode. If the file contains incorrect entries (too many/few columns, incorrect mode) the command will report these but will not execute any of the entries.

-b *batch size*

If the input comes from a file, the command will process all entries and send the request to the server in chunks specified by this option. The default is 1000 entries at a time.

`-s csv formatted line`

Supply a single CSV formatted line. Eliminates the need to create a file for a simple change

`-jndiHost hostname`

The server jndi host; the default is localhost

`-jndiPort port`

The server jndi port; the default is 1099

`-u <username>`

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-p <password>`

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

## Syntax of Comma Separated File for node management mode

The CSV file you supply must have the following syntax for setting management mode on nodes.

Empty lines are ignored.

Lines that begin with the # character are ignored.

All columns are mandatory

- Column 1(A) : Node DNS|IP Address

Specify the DNS name of the node OR the IP address. The search algorithm is as follows:

- find a node that has a matching management ip address
- find a node that has a matching ip address in the default domain
- find a node that has a matching host name
- find a node that has a matching name

- Column 2(B) : Management mode

The mode to set the node to. Valid entries are "MANAGED", "NOTMANAGED", "OUTOFSERVICE"

Example lines:

192.168.1.1,OUTOFSERVICE

my.fqdn.com, MANAGED

## Syntax of Comma Separated File for interface management mode

The CSV file you supply must have the following syntax for setting management mode on interfaces.

Empty lines are ignored.

Lines that begin with the # character are ignored.

All columns are mandatory

- Column 1(A) : Node DNS|IP Address

Specify the DNS name of the node OR the IP address. The search algorithm is the same as specified above.

- Column 2(B) : Interface id

Interface Id Specify the identifier of the interface on the node specified in the previous field. The search algorithm is as follows:

- ifIndex
- ifName
- ifAlias
- ifDescription

- Column 3(C) : Mode

The management mode to set the interface to. Valid entries are "INHERITED", "NOTMANAGED", "OUTOFSERVICE"

Example lines:

```
192.168.1.1,1,OUTOFSERVICE
```

```
my.fqdn.com, myAlias, OUTOFSERVICE
```

## RETURN VALUE

`nnmmanagementmode.ovpl` exits with the status 0 (zero) if no errors were encountered, 1 otherwise.

## AUTHOR

`nnmmanagementmode.ovpl` was developed by Hewlett-Packard Company.

## EXTERNAL INFLUENCES

### Environmental Variables

[Return to Reference Pages Index](#)

## Name

`nnmnetdeletenodeattrs.ovpl` — script to delete node custom attributes from a comma separated values (CSV) file

## SYNOPSIS

```
nnmnetdeletenodeattrs.ovpl [-?] [-u <username> -p <password>] [-f <csv_filename>] [-s <csv formatted line>]
```

## DESCRIPTION

**IMPORTANT:** This command is deprecated and will be removed in a future release. Please use `nnmdeleteattributes.ovpl` instead.

`nnmnetdeletenodeattrs.ovpl` allows node custom attributes to be deleted from a comma separated values (CSV) file, such as a .csv file from Microsoft™ Excel. This command is useful if you have previously created custom attributes that are no longer needed. Once the attributes are deleted, any nodes in a node group formed by referencing those attributes will disappear from the node group.

**NOTE:** unlike `nnmnetloadnodeattrs.ovpl` this command does NOT require a valid iSPI-NET license.

You must be logged on as root/administrator to run this command.

## Parameters

`nnmnetdeletenodeattrs.ovpl` supports the following options:

- ?  
Prints the usage statement.
- u <username>  
Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.
- p <password>  
Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.
- f <csv\_filename>  
Supply the CSV file name (with path, e.g. /opt/tmp/mynodes.csv) that lists the custom attributes to delete.
- s <csv formatted line>

Supply a single CSV formatted line. Eliminates the need to create a file for a simple change

## Syntax of Comma Separated File

The CSV file you supply must have the following syntax for adding attributes.

Empty lines are ignored.

Lines that begin with the # character are ignored.

- Column 1(A) : Node DNS|IP Address

Specify the DNS name of the node OR the IP address. This field is compulsory.

- Column 2(B) : Attribute Name

The name of the custom attribute.

Additional attribute names may be specified on the same line or on a separate line with the same node DNS|IP Address.

Example lines:

```
192.168.1.1,Project,Service Type
```

```
192.168.1.1,Asset Tracking
```

```
192.168.2.2,Project,Service Type,Asset Tracking
```

## Use of Microsoft Excel

Microsoft Excel is a handy tool to create comma separated files, but .csv files do not maintain the Excel spreadsheet's column width, comments, etc. It is recommended that you store a nnmnetdeletenodeattrs.ovpl input file as a native .xls format, and then perform `File:Save As...` to create a .csv file. Then you can add Excel comments to the file, make columns wider, and you do not need to worry about escaping the comma character.

## EXAMPLES

Sample CSV file contents:

```
192.168.2.2,Project,Service Type,Asset Tracking
```

To delete the Node custom attributes from a CSV file:

```
nnmnetloadnodeattrs.ovpl -u system -p myadminpasswd -f /tmp/test.csv
```

To delete the Node custom attributes from the command line:

```
nnmnetloadnodeattrs.ovpl -u system -p myadminpasswd -s "192.168.1.1,Project"
```

## AUTHOR

`nnmnetloadnodeattrs.ovpl` was developed by Hewlett-Packard Company.

## FILES

`$NNM_BIN/nnmnetloadnodeattrs.ovpl`

## SEE ALSO

[nnmnetloadnodeattrs.ovpl\(1M\)](#), [nnmloadnodegroups.ovpl\(1M\)](#), [nnm.properties\(4\)](#).

[Return to Reference Pages Index](#)

## Name

`nnmnetloadnodeattrs.ovpl` — script to load node custom attributes from a comma separated values (CSV) file

## SYNOPSIS

```
nnmnetloadnodeattrs.ovpl [-?] [-u <username> -p <password>] [-f <csv_filename>] [-r <true / false>] [-s <csv formatted line>]
```

## DESCRIPTION

**IMPORTANT:** This command is deprecated and will be removed in a future release. Please use `nnmloadattributes.ovpl` instead.

`nnmnetloadnodeattrs.ovpl` allows node custom attributes to be loaded from a comma separated values (CSV) file, such as a .csv file from Microsoft™ Excel. This command is useful if you have a large number of nodes that are defined in an external datastore, and you would like to load these attributes into NNM. After loading into NNM you can use the Node Group forms to group nodes according to their custom attributes.

NOTE: this utility requires a valid iSPI-NET license.

You must be logged on as root/administrator to run this command.

## Parameters

`nnmnetloadnodeattrs.ovpl` supports the following options:

`-?`

Prints the usage statement.

`-u <username>`

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-p <password>`

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-r <true | false>`

Existing attribute values are not changed unless this argument is provided with the value of true.

`-f <csv_filename>`

Supply the CSV file name (with path, e.g. /opt/tmp/mynodes.csv) from where you want to import the custom attributes.

```
-s <csv formatted line>
```

Supply a single CSV formatted line. Eliminates the need to create a file for a simple change

## Syntax of Comma Separated File

The CSV file you supply must have the following syntax for adding attributes.

Empty lines are ignored.

Lines that begin with the # character are ignored.

- Column 1(A) : Node DNS|IP Address

Specify the DNS name of the node OR the IP address. This field is compulsory.

- Column 2(B) : Attribute Name

The name of the custom attribute.

- Column 3(C) : Attribute Value

The value of the custom attribute.

Additional attribute name and value pairs may be specified on the same line or on a separate line with the same node DNS|IP Address.

Example lines:

```
192.168.1.1,Location,Building Five Upper,Service Type,eCommerce
```

```
192.168.1.1,Asset Tracking,N1234
```

```
192.168.2.2,Location,Fort Collins,Service Type,IT,Asset Tracking,F4321
```

## Use of Microsoft Excel

Microsoft Excel is a handy tool to create comma separated files, but .csv files do not maintain the Excel spreadsheet's column width, comments, etc. It is recommended that you store a nnmnetloadnodeattrs.ovpl input file as a native .xls format, and then perform `File:Save As...` to create a .csv file. Then you can add Excel comments to the file, make columns wider, and you do not need to worry about escaping the comma character.

## EXAMPLES

Sample CSV file contents:

```
192.168.2.2,Location,Fort Collins,Service Type,IT,Asset Tracking,F4321
```

To load the Node custom attributes from a CSV file overwriting existing values:

```
nnmnetloadnodeattrs.ovpl -u system -p myadminpasswd -f /tmp/test.csv -r true
```

To load the Node custom attributes from the command line:

```
nnmnetloadnodeattrs.ovpl -u system -p myadminpasswd -s "192.168.1.1,Project,IT Update of Building Five"
```

## AUTHOR

nnmnetloadnodeattrs.ovpl was developed by Hewlett-Packard Company.

## FILES

`$NNM_BIN/nnmnetloadnodeattrs.ovpl`

## SEE ALSO

[nnmnetdeletenodeattrs.ovpl\(1M\)](#), [nnmloadnodegroups.ovpl\(1M\)](#), [nnm.properties\(4\)](#).

[Return to Reference Pages Index](#)

## Name

nnmnodedelete.ovpl — Remove node(s) and associated data from the NNM topology database

## SYNOPSIS

```
nnmnodedelete.ovpl -help | -node <hostName> | -rm <Regional NNMi management server> | -file  
<filename> [-u <username> -p <password>] [-jndiHost <hostName> Default: localhost] [-jndiPort  
<port> Default: 1099]
```

## DESCRIPTION

nnmnodedelete.ovpl removes a node and its associate data like interfaces, ip addresses etc from the system. If this results in empty VLANs and/or subnets they will be removed as well. If incidents point to this node, the Source Node field will be blanked out, but the incidents will not be removed. The node is identified using the hostName field.

The -rm option accepts the name of a Regional NNMi management server. Nodes that are managed by that Regional Manager, will be removed from the local database.

The -file option accepts a file with a single entry specified per line. Each line has the following format:

*HostName* # (optional comment to help identify the node, if desired) Where: *HostName* = the host name of the node you wish to add. Comments can be delimited with a # character.

## Parameters

nnmnodedelete.ovpl supports the following options:

-help

Prints the usage statement.

-node <hostName>

The hostname of the node to delete.

-rm <RegionalManagerName>

The name of the Regional NNMi management server.

-file <fileName>

Specify a text file to read the nodes from.

-u <username>

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

-p <password>

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

-jndiHost <serverName>

The server JNDI host; default is localhost.

-jndiPort <port>

The server JNDI port; default is 1099.

## EXAMPLES

```
nnmnodedelete.ovpl -u username -p password -node myNode
```

Removes the node `myNode`. (You must provide an NNM username and password.)

```
nnmnodedelete.ovpl -u username -p password -rm myRegionalManager
```

Removes all nodes associated with `myRegionalManager` (You must provide an NNM username and password.)

```
nnmnodedelete.ovpl -u username -p password -file myFile
```

Reads the nodes specified in the `myFile` file and attempts to remove them from the database. (You must provide an NNM username and password.)

## Diagnostics

`nnmnodedelete.ovpl` returns the following exit codes:

0

Operation was successful.

1

An error occurred; see error message for details.

2

Partly successful but some nodes were not deleted; see error messages for details.

## AUTHOR

`nnmnodedelete.ovpl` was developed by Hewlett-Packard Company.

## FILES

`$NNM BIN/nnmnodedelete.ovpl`

## SEE ALSO

## NOTES

The deleted node could be rediscovered if it was created due to a Auto Discovery Rule. The workaround would be to add the node's ip address to the Excluded IP Addresses entry in the Discovery Configuration form.

[Return to Reference Pages Index](#)

## Name

`nnmmoderediscover.ovpl` — Re-discover nodes

## SYNOPSIS

```
nnmmoderediscover.ovpl -help | -node <hostName> | -rm <Regional NNMi management server> | -file  
<filename> | -all [-u <username> -p <password>] [-jndiHost <hostName> Default: localhost] [-jndiPort  
<port> Default: 1099]
```

## DESCRIPTION

`nnmmoderediscover.ovpl` Injects node(s) into the NNMi discovery queue. The amount of time before the node starts discovery is dependent on how long NNMi takes to get to the node in the queue.

Note that if the node is already in the discovery queue it is not added again.

Note that when using the NNMi Advanced – Global Network Management feature, nodes that are managed by a Regional Manager go through discovery on the Regional Manager (NNMi management station) only and are not rediscovered by the Global Manager.

The `-rm` option is for the NNMi Advanced – Global Network Management feature when issuing the `nnmmoderediscover.ovpl` command on a Global Manager. NNMi sends a request to the Regional Manager requesting that the Regional Manager (NNMi management server) send the most recent available discovery results to the Global Manager.

The `-file` option accepts a file with a single entry specified per line. Each line contains a short name, or a fully-qualified DNS domain name, or an IP address. Each line has the following format: `HostName #` (optional comment to help identify the node, if desired) Where: `HostName` = the short name or DNS name or IP address of the node you wish to add.

The `-all` option causes all nodes managed by the local NNMi management server to be rediscovered. Note when using the NNMi Advanced – Global Network Management feature and issuing the `nnmmoderediscover.ovpl` command on a Global Manager, nodes that are managed by a Regional Manager have the most recent available discovery results sent to the Global Manager from the Regional Manager.

## Parameters

`nnmmoderediscover.ovpl` supports the following options:

`-help`

Prints the usage statement.

`-node <hostName>`

The hostname of the node to rediscover.

When using the NNMi Advanced - Global Network Management feature and issuing the `nnmnode rediscover.ovpl` command on a Global Manager, the Name attribute value from a Regional Manager Connection configuration (that identifies a Regional NNMi management server).

-file <fileName>

Specify a text file to read the nodes from.

-all

Specify all nodes are to be re-discovered.

-u <username>

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

-p <password>

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

-jndiHost <serverName>

The server JNDI host; default is localhost.

-jndiPort <port>

The server JNDI port; default is 1099.

## EXAMPLES

```
nnmnode rediscover.ovpl -u username -p password -node myNode
```

Re-discovers the node `myNode`. (You must provide an NNM username and password.)

```
nnmnode rediscover.ovpl -u username -p password -rm myRegionalManager
```

Re-discovers all nodes associated with `myRegionalManager` (You must provide an NNM username and password.)

```
nnmnode rediscover.ovpl -u username -p password -file myFile
```

Reads the nodes specified in the `myFile` file and injects them into the NNMi discovery queue. (You must provide an NNM username and password.)

## Diagnostics

`nnmnode rediscover.ovpl` returns the following exit codes:

0

Operation was successful.

An error occurred; see error message for details.

## **AUTHOR**

`nnmmoderediscover.ovpl` was developed by Hewlett-Packard Company.

## **FILES**

`$NNM_BIN/nnmmoderediscover.ovpl`

## **SEE ALSO**

## **NOTES**

[Return to Reference Pages Index](#)

## Name

`nnmofficialfqdn.ovpl` — Display the official Fully Qualified Domain Name (FQDN) of the NNMi management server.

## SYNOPSIS

`nnmofficialfqdn.ovpl`

## DESCRIPTION

Use `nnmofficialfqdn.ovpl` to display the official Fully Qualified Domain Name (FQDN). The official FQDN enables Single Sign On with NNM iSPiS. NNMi sets the official FQDN during installation. After installation, you can change it using the `nnmsetofficialfqdn.ovpl` command.

## Options

`nnmofficialfqdn.ovpl` by default displays verbose output with a ping test. `nnmofficialfqdn.ovpl` recognizes the following options:

`-d`

Display the domain name of the official FQDN. If the FQDN is only a short hostname or an IP address and if the terse option (`-t`) is used, NNMi does not display a value; otherwise, NNMi displays a message informing you that the domain name cannot be found.

`-t`

Terse mode. Displays the FQDN or domain name value only. When the FQDN or domain name is not found, no warning or informative text is shown. This option also skips the ping test.

`-m`

Query for the default FQDN and display the value. NNMi displays whichever value it finds first using the following order:

- FQDN
- Short hostname
- IP address

If none of the above are found, then 'localhost' is displayed.

`-h`

Display the help menu listing all the options.

## EXAMPLES

Running the command without any options displays the official FQDN and performs a ping test:

```
# nnmofficialfqdn.ovpl
  FQDN: hostname.somedomain
  Ping test: pinging hostname.somedomain please wait...
  Ping OK
```

Running the command with the -t option displays the official FQDN as follows:

```
# nnmofficialfqdn.ovpl -t
  hostname.somedomain
```

Running the command using the -d option displays the domain name as follows:

```
# nnmofficialfqdn.ovpl -d
  Domain: somedomain
```

Running the command using the -d and -t options displays only the domain name:

```
# nnmofficialfqdn.ovpl -dt
  somedomain
```

## AUTHOR

nnmofficialfqdn.ovpl was developed by Hewlett-Packard Company.

## FILES

nnmofficialfqdn.ovpl resides in \$NNM\_BIN directory.

## SEE ALSO

[nnmsetofficialfqdn.ovpl\(1M\)](#), [nnmsetdomain.ovpl\(1M\)](#), [lwssofmconf.xml\(4\)](#).

[Return to Reference Pages Index](#)

## Name

`nnmopcxport.ovpl` — export NNMi management event and SNMP trap configurations to HP OM policies (SNMP template) file.

## SYNOPSIS

```
nnmopcxport.ovpl -help
```

```
nnmopcxport.ovpl [-u <username> -p <password>] [-jndiHost <hostname> -jndiPort <port>] -template <template_name> -application <application_name> [-agtmsi_copy | -agtmsi_divert] [-svmsi_copy | -svmsi_divert] [-msgtype] [-author <author code>] [-oid <snmp oid prefix>] [-omi_hi] -file <output_file name> [-force]
```

```
nnmopcxport.ovpl [-u <username> -p <password>] [-jndiHost <hostname> -jndiPort <port>] -template <template_name> -application <application_name> [-agtmsi_copy | -agtmsi_divert] [-svmsi_copy | -svmsi_divert] [-msgtype] [-author <author code>] [-oid <snmp oid prefix>] [-omi_hi] -omi_policy
```

## DESCRIPTION

`nnmopcxport.ovpl` reads the NNMi management event and SNMP trap configurations and exports the configurations as an HP OM policies file. This tool provides similar functionality to the HP OM `ovtrap2opc` utility that converted legacy NNM `trapd.conf` files into HP OM policies.

## Parameters

`nnmopcxport.ovpl` supports the following parameters:

`-help`

Show help.

`-u <username>`

Supply the NNMi administrator username required to execute the command. Required unless the user has root or administrator privileges or a [nnm.properties\(4\)](#) file exists.

`-p <password>`

Supply the NNMi administrator password required to execute the command. Required unless the user has root or administrator privileges or a [nnm.properties\(4\)](#) file exists.

`-agtmsi_copy`

Copy matching messages to the OVO Agent Message Stream Interface.

`-agtmsi_divert`

Divert matching messages to the OVO Agent Message Stream Interface.

`-application <application_name>`

Set the policy application name.

`-author <author_code>`

Export events matching the author code.

`-file <output_filename>`

Save output to file with the specified name.

`-force`

Overwrite an existing policy file specified by `-file <output_file_name>`.

`-jndiHost <hostname>`

The server jndi host; default is localhost.

`-jndiPort <port>`

The server jndi port; default is 1099.

`-msgtype`

Set the message type to the name of the condition. The value of the resulting MSGTYPE field will be truncated to 32 characters for OM import compatibility.

`-oid <snmp_oid_prefix>`

Export events with snmp trap OIDs matching the provided OID.

`-omi_hi`

Include HP OMi health indicators for applicable NNMi management events.

`-omi_policy`

Export in OMi policy format. A header file and a data file are created of the form `<UUID>_header.xml` and `<UUID>_data` where UUID is an Universally Unique Identifier. The `-file` option is ignored.

`-svmsi_copy`

Copy matching messages to the OVO Server Message Stream Interface.

`-svmsi_divert`

Divert matching messages to the OVO Server Message Stream Interface.

`-template template name`

Set the `<template_name>`.

## EXAMPLES

Export an OM policy file containing all NNMi management events and SNMP trap definitions:

```
nnmopcexport.ovpl -u user -p pass -template "SNMP Traps" -application "NNMi" -file policies.dat
```

Export an OMi policy file containing all NNMi management events and SNMP trap definitions and OMi health indicators:

```
nnmopcexport.ovpl -u user -p pass -template "SNMP Traps" -application "NNMi" -omi_hi -omi_policy
```

## AUTHOR

nnmopcexport.ovpl was developed by Hewlett-Packard Company.

[Return to Reference Pages Index](#)

## Name

`nnmprincipalconfig.ovpl` — script to add or delete users in NNMi

## SYNOPSIS

```
nnmprincipalconfig.ovpl [-?] -add | -delete -username <new_user_name> -password  
<new_user_password> -role <new_user_role> [-u <username> -p <password>]
```

## DESCRIPTION

`nnmprincipalconfig.ovpl` is a command line tool that adds, deletes users for NNMi or modifies existing user's password and/or role assignments. This command should not be used if managing users and/or roles via an LDAP or active directory service.

You must be logged on as `root` user or `NNMi administrator` to run this command.

## Parameters

`nnmprincipalconfig.ovpl` supports the following options:

- ?  
Prints the usage statement.
- add  
Add the specified NNMi user with the specified role.
- delete  
Delete the specified NNMi user.
- username  
The name of the user to be added or deleted.
- password  
The password of the new user being added.
- role  
The role of the new user being added. Must be one of: `admin`, `level2`, `level1`, `guest`, or `client`.
- u  
Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.
- p

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

[Return to Reference Pages Index](#)

## Name

`nnmresetembdb.ovpl` — drop (delete) and recreate the embedded database, if NNMi is configured to run with the embedded database

## SYNOPSIS

```
nnmresetembdb.ovpl [-?|-h|-help] [-silent] [-nostart]
```

## DESCRIPTION

Use `nnmresetembdb.ovpl` to drop (delete) and recreate the NNMi embedded database. This command is useful only if you installed NNMi with the embedded database option. You should use this command only if your database is corrupt, and you are willing to lose all of your data; or, if you simply want to reset your database to the state it was in after initial installation.

If NNMi is running when this command is executed, the command first stops NNMi (using `ovstop`), then starts the `nmsdbmgr` process to work with the database (using `ovstart`). Unless the `-nostart` option is present, NNMi restarts (using `ovstart`) upon successful completion of the reset process.

When the database reset has completed, the embedded database has no tables or data in it. The tables are recreated when you restart NNMi using the `ovstart` command, or when the `nnmresetembdb.ovpl` command automatically starts NNMi. You must be logged in as `root` on UNIX systems, or as `administrator` on Windows systems to run this command.

## Parameters

The `nnmresetembdb.ovpl` command recognizes the following parameters:

`-silent`

The `nnmresetembdb.ovpl` command does not display the command results when you use the `-silent` option.

`-nostart`

The `nnmresetembdb.ovpl` command does not start NNMi after the database reset when you use the `-nostart` option.

`-?|-h|-help`

Display command usage.

## EXAMPLES

You can use this script to reset your database if you have a bad discovery or a corrupted database; or if you

want to reset the database (including configuration items stored in the database) to the state the database was in after initial installation.

You'll see these messages:

```
# nnmresetembdb.ovpl -nostart
```

```
WARNING: Running this tool will stop NNM, drop and recreate the database,  
and restart NNM. Please make sure no major activity is occurring at  
this time.
```

```
Is it OK to stop NNM (ovstop)? [n] y  
Thank you!
```

```
WARNING: This will delete all configuration and discovered data. You cannot  
recover from a reset unless you have taken a backup.
```

```
Are you sure you wish to reset your database? [n] y
```

```
Attempting to reset the embedded database...  
Starting nmsdbmgr process for database reset...  
Successfully started nmsdbmgr process for database reset.  
Successfully reset private database.  
Successfully reset NNM embedded database.
```

```
#
```

## AUTHOR

nnmresetembdb.ovpl was developed by Hewlett-Packard Company.

## FILES

nnmresetembdb.ovpl resides in `$NNM_BIN` (UNIX) or `install_dir\bin` (Windows) directories.

## SEE ALSO

ovstart(1M), ovstop(1M), ovstatus(1M), nmsdbmgr(1M).

[Return to Reference Pages Index](#)

## Name

`nnmrestore.ovpl` — restore a backup created by the `nnmbackup.ovpl` script

## SYNOPSIS

```
nnmrestore.ovpl [-?|-h|-help] [-force] [-lic] [-partial] -source <directory>
```

## DESCRIPTION

`nnmrestore.ovpl` serves as the main restore script for NNMi. The command uses a previous NNMi backup performed with the `nnmbackup.ovpl` script to restore NNMi to the state stored in the backup files. The scope of the restore is determined by the content of the backup and by command-line arguments you provide. Only data present in the backup is restored.

The command detects the following:

- Was the backup performed on the target system? If the backup was taken locally, the `-lic` parameter allows for the restoration of licensing information.
- Is the target system running an Oracle or Postgres database? If you require a full restore of an Oracle database, i.e. a `nnmrestore.ovpl` without the (`-partial`) parameter, you will be prompted to determine: if the target system is using either the same Oracle database instance as the backup, or if an Oracle backup and restore operation has already taken place on the target system. You must answer these questions correctly or SSL certificates will be copied or merged incorrectly and the database will become inaccessible from the target installation.

The source directory contains all of the files required for the restore options you have selected, or a single tar file. If the source is a tar file, the tar file is extracted to a temporary folder in the current working directory. The temporary folder is removed after the restore is complete.

NNMi must be stopped to complete restore operations. If you use the `-force` option, the script stops NNMi. If the files present in the source folder indicate that the original backup was an `online` backup, the restore process starts the `nmsdbmgr` process to make sure the embedded database is available. To restore from an `online` backup, the `-force` option **MUST** be used.

You must be logged in as `root` on UNIX systems or as `administrator` on Windows systems to run this command.

### Caution

- `nnmrestore.ovpl` can restore backups taken with obsolete versions of `nnmrestore.ovpl`; however, while the command allows you to restore a backup taken from a different system, the certificate merge does not occur and the restore corrupts the target installation.
- The database restore performed by this command applies only to the embedded database. If you chose a different database at installation, your table data is not

restored using this command. File-system restores work regardless of database type,. For details about what you need to restore if you select a different database at installation, refer to the *NNMi Deployment Guide* (available at <http://h20230.www2.hp.com/selfsolve/manuals>).

## Parameters

`nnmrestore.ovpl` supports the following options:

`-force`

If you provide this option, the command stops NNMi before running the restore procedure. Also, the command starts the `nmsdbmgr` process if it is required to restore the type of backup on which the restore is based. This option is required to restore from an online backup.

`-lic`

If you provide this option, the command restores licensing information. Note that the command only restores licensing information if it successfully validates that the backup operation was performed on the same system.

`-partial`

Without this option, the database and corresponding SSL certificates will be restored. In the case of a system-to-system restore, the `nnm.keystore` and `nnm.truststore` are merged with the target system's. The merge routine merges all certificate aliases from the backup stores that do not already exist on the target system's. There is one exception, if the `FQDN.selfsigned` key alias exists in both stores, the target system's key alias is removed and replaced with the one stored in the backup. Before the certificates are merged or restored, a backup is performed, stored in the same directory, and labeled `*.original`.

With the `-partial` option, the database and corresponding SSL certificates will not be restored. This is useful for restoring configuration files only.

`-source <directory>`

Specifies the input directory where the restore files are stored. If the source you provide is a tar file, the tar file is extracted to a temporary folder in the current working directory. The temporary folder is removed when the restore is complete.

`-?|-h|-help`

Display command usage.

## EXAMPLES

To restore a previous backup:

```
#!/nmrestore.ovpl -source /tmp/bak/config
```

To restore with the `-force` option:

```
#!/nmrestore.ovpl -force -source /tmp/bak/all
```

To restore everything except the database, SSL certificates, and licenses, use the `-partial` option:

```
#!/nmrestore.ovpl -partial -source /tmp/bak/all
```

To restore licensing information on a local system use the `-lic` option:

```
#!/nmrestore.ovpl -lic -source /tmp/bak/all
```

## AUTHOR

`nmrestore.ovpl` was developed by Hewlett-Packard Company.

## SEE ALSO

`nnmbackup.ovpl(1M)`.

[Return to Reference Pages Index](#)

## Name

`nnmrestoreembdb.ovpl` — restore a full backup of the NNMi embedded database

## SYNOPSIS

```
nnmrestoreembdb.ovpl [-?|-h|-help] [-force] -source <backup file>
```

## DESCRIPTION

Use `nnmrestoreembdb.ovpl` to restore a full backup of the NNMi embedded database. You create the backup file required for the restore using the `nnmbackupembdb.ovpl` command. Because query planning statistics are not backed up, these statistics are gathered as part of the restore procedure to ensure good database performance after a restore.

Before running this command, make sure the embedded database is empty. You can do this by running the `nnmresetembdb.ovpl` command. If you do not clear the embedded database, the restore procedure runs this command for you and fails if the reset command fails.

Do *not* run this command while NNMi is running unless you specify the `-force` option. Only the `nmsdbmgr` process can (and must) be running when this command is executed. You must be logged in as root on UNIX systems, or as administrator on Windows systems to run this command.

## Parameters

`-source <backup file>`

File from which to restore the backup. This file must be a file that was created using the `nnmbackupembdb.ovpl` command.

`-force`

If you provide this option, the command stops NNMi if it is currently running, and then starts the `nmsdbmgr` process.

`-?|-h|-help`

Display command usage.

## EXAMPLES

You can use this script to run a full database recovery when a restore from backup is necessary.

You'll see these messages:

```
# nnmrestoreembdb.ovpl -source /backups/nnmDb.dump
```

```
WARNING: Running this command while NNMi is running will cause sporadic
```

```
failures in active sessions. Please ensure that only the nmsdbmgr
process is running when executing this command (ovstart nmsdbmgr).
Are you sure you want to run a full database restore now? [n] y

Ok, performing full embedded database restore...
Statistics analysis completed successfully.

NNMi embedded database successfully restored from /backups/nnmDb.dump.
#
```

## AUTHOR

nnmrestoreembdb.ovpl was developed by Hewlett-Packard Company.

## FILES

nnmrestoreembdb.ovpl resides in the following directories:

- UNIX: /opt/OV/bin
- Windows: %NnmInstallDir%\bin.

## SEE ALSO

ovstart(1M), ovstop(1M), ovstatus(1M), nmsdbmgr(1M), nnmbackupembdb.ovpl(1M).

[Return to Reference Pages Index](#)

## Name

`nnmseeddelete.ovpl` — Remove seed from the NNM topology database

## SYNOPSIS

```
nnmseeddelete.ovpl -help | -seed <seed> | [-u <username> -p <password>] [-jndiHost <hostName>  
Default: localhost] [-jndiPort <port> Default: 1099]
```

## DESCRIPTION

`nnmseeddelete.ovpl` removes a seed from the system.

## Parameters

`nnmseeddelete.ovpl` supports the following options:

`-help`

Prints the usage statement.

`-seed <seed>`

The seed to delete. A seed can be a host name or an IP address and must match exactly as listed in the seed list.

`-u <username>`

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-p <password>`

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-jndiHost <serverName>`

The server JNDI host; default is `localhost`.

`-jndiPort <port>`

The server JNDI port; default is `1099`.

## EXAMPLES

```
nnmseeddelete.ovpl -u username -p password -seed 10.1.1.2.3
```

Removes the seed 10.1.2.3. (You must provide an NNM username and password.)

## Diagnostics

`nmseeddelete.ovpl` returns the following exit codes:

0

Operation was successful.

1

An error occurred; see error message for details.

## AUTHOR

`nmseeddelete.ovpl` was developed by Hewlett-Packard Company.

## FILES

`$NNM_BIN/nmseeddelete.ovpl`

## SEE ALSO

[Return to Reference Pages Index](#)

## Name

`nnmsetdampenedinterval.ovpl` — sets the dampened interval for all incident configurations

## SYNOPSIS

```
nnmsetdampenedinterval.ovpl [ [-hours hours] [-minutes minutes] [-seconds seconds] [-u username] [-p password] ]
```

## DESCRIPTION

A *dampening interval* is the amount of time NNMi holds an incident in a dampened, or delayed, state. After the dampening interval passes, NNMi changes the state of an incident to `Registered`, then acts upon the incident using actions or correlations. If NNMi closes an incident during the dampened interval, it also deletes the incident. The `nnmsetdampenedinterval.ovpl` script sets the dampened interval for all incident configurations. The maximum dampened interval you can set is 60 minutes. HP recommends a dampened interval of at least 6 minutes. Specify a value for at least one of the following variables: hours, minutes, or seconds. To disable dampening, set the values for hours, minutes, and seconds to 0. Run the `nnmsetdampenedinterval.ovpl` script after installing NNMi 9.0 to ease migration from NNMi 8.13.

## Parameters

`-hours hours`

Specify the number of hours for the dampened interval. If specified, the value for *hours* must be greater than or equal to 0.

`-minutes minutes`

Specify the number of minutes for the dampened interval. If specified, the value for *minutes* must be greater than or equal to 0.

`-seconds seconds`

Specify the number of seconds for the dampened interval. If specified, the value for *seconds* must be greater than or equal to 0.

`-jndiHost hostname`

Server jndi host. Default is `localhost`.

`-jndiPort port`

Server jndi port. Default is 1099.

`-u <username>`

Supply the NNMi administrator username required to execute the command. Required unless a

[nnm.properties\(4\)](#) file exists.

-p <password>

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

## EXAMPLES

Set the dampened interval for all incident configurations to 10 minutes 10 seconds:

```
nnmsetdampenedinterval.ovpl -hours 0 -minutes 10 -seconds 10
```

Set the dampened interval for all incident configurations to 1 hour:

```
nnmsetdampenedinterval.ovpl -hours 1
```

Set the dampened interval for all incident configurations to 6 minutes:

```
nnmsetdampenedinterval.ovpl -minutes 6
```

Set the dampened interval for all incident configurations to 30 seconds:

```
nnmsetdampenedinterval.ovpl -seconds 30
```

Set the dampened interval for all incident configurations to 10 minutes 10 seconds:

```
nnmsetdampenedinterval.ovpl -minutes 10 -seconds 10
```

Disable dampening for all incident configurations:

```
nnmsetdampenedinterval.ovpl -hours 0 -minutes 0 -seconds 0
```

## AUTHOR

nnmsetdampenedinterval.ovpl was developed by Hewlett-Packard Company.

## SEE ALSO

[nnm.properties\(4\)](#)

[Return to Reference Pages Index](#)

## Name

`nnmsetdomain.ovpl` — Set the domain for SSO, in the SSO configuration file.

## SYNOPSIS

```
nnmsetdomain.ovpl [-h] [-a|r] [-m]
```

## DESCRIPTION

`nnmsetdomain.ovpl` sets the domain name in `$NnmDataDir/shared/nnm/conf/lwssofmconf.xml` or if `-a` | `-r` is provided as a target xml configuration.

You must be logged in as `root` on UNIX systems or as `administrator` on Windows systems to run this command.

## Parameters

`nnmsetdomain.ovpl` supports the following options:

- `-a`  
Specify an absolute path to conf file. If no path is provided the default SSO configuration file will be used. Path names should be enclosed in quotes for Windows.
- `-r`  
Specify path relative to `$NnmDataDir`. If no path is provided the default SSO configuration file will be used.
- `-m`  
When this option is used, a manual lookup for the domain is performed ONLY IF domain lookup from the predefined FQDN value fails.
- `-h`  
Display command usage.

## EXAMPLES

To set the domain in the default SSO configuration file:

```
#!/nnmsetdomain.ovpl
```

To set the domain in the default configuration file and lookup the domain if necessary:

```
#!/nnmsetdomain.ovpl -m
```

To set the domain in a targeted configuration file:

```
#./nmmsetdomain.ovpl -a /opt/spi/framerelay/server/nnm/conf/lwssconf.xml
```

## AUTHOR

nmmsetdomain.ovpl was developed by Hewlett-Packard Company.

## SEE ALSO

[nnmofficialfqdn.ovpl\(1M\)](#), [nmmsetofficialfqdn.ovpl\(1M\)](#), [lwssofmconf.xml\(4\)](#).

[Return to Reference Pages Index](#)

## Name

`nnmsetofficialfqdn.ovpl` — Set the official Fully Qualified Domain Name (FQDN) of the NNMi management server.

## SYNOPSIS

```
nnmsetofficialfqdn.ovpl [-f | -force] <fqdn>
```

## DESCRIPTION

Use `nnmsetofficialfqdn.ovpl` to change the official FQDN for the NNMi management server after installation. When run without any arguments, it sets the official FQDN to the default value. The default value is obtained by performing a hostname lookup. See the reference page for the `nnmofficialfqdn.ovpl` command for more information on default hostname lookup using the `-m` option.

If the FQDN is changed the user is prompted to auto generate a new SSL certificate. All certificates that NNMi auto generates have the following alias syntax: `fqdn.selfsigned`. Because the SSL certificate is tied to the FQDN, HTTPS will be broken if a new certificate is not generated and added to both the keystore and truststore. If `-force` is provided then the certificate will automatically be generated and inserted into both the `nm.keystore` and `nm.truststore`. If the new certificate's alias already exists in either the keystore or truststore it is not inserted.

If a new certificate is generated the system property `com.hp.ov.nms.ssl.KEY_ALIAS` in `nms-local.properties` is updated to reference the new certificate alias. This is a necessary step for HTTPS to work correctly when NNMi is started again.

## Options

There are two optional arguments to `nnmsetofficialfqdn.ovpl`

`<fqdn>`

Resets the official FQDN to `<fqdn>` upon confirmation from the user.

If the provided value of `<fqdn>` does not have a domain (for example, it is a short hostname, IP address or localhost), Single Sign On is disabled.

`-force | -f`

This is a "force" flag that does not prompt the user for confirmation and does not print verbose informational text. When used alone it sets the official FQDN to the default. When `<fqdn>` is used after this flag, it changes the official FQDN to `<fqdn>`.

## EXAMPLES

Set the FQDN to the default value:

```
# nmmsetofficialfqdn.ovpl
```

Set the FQDN to the value somehost.somedomain:

```
# nmmsetofficialfqdn.ovpl somehost.somedomain
```

Force the official FQDN to the default value:

```
# nmmsetofficialfqdn.ovpl -f
```

## AUTHOR

nmmsetofficialfqdn.ovpl was developed by Hewlett-Packard Company.

## FILES

nmmsetofficialfqdn.ovpl resides in \$NNM\_BIN directory.

## SEE ALSO

[nmmofficialfqdn.ovpl\(1M\)](#), [nmmsetdomain.ovpl\(1M\)](#), [lwssofmconf.xml\(4\)](#).

[Return to Reference Pages Index](#)

## Name

nnmsnmpbulk.ovpl — query a node for information using an SNMPv2c GetBulk request

## SYNOPSIS

```
nnmsnmpbulk.ovpl -u user_name -p passwd [options] node object-id [,object-id]...
```

*options*: [-d] [-v *version*] [-c *community*] [-port *port*(default:161)] [-t *timeout*(default:5000)] [-r *retries*(default:1)] [-T] [-n *non-repeaters*] [-m *max-repetitions*] [-pp *Proxy Port*] [-pa *Proxy Address*] [-a *Authentication Protocol*] [-A *Authentication Pass phrase*] [-X *Privacy Protocol*] [-X *Privacy Passphrase*] [-N *Context Name*] [-v3u *SNMPv3 user name*]

## DESCRIPTION

nnmsnmpbulk.ovpl uses the SNMPv2c/v3 GetBulk request to retrieve information from an SNMP agent. The SNMP GetBulk request minimizes the number of protocol exchanges required to retrieve a large amount of information. Performance is increased, as fewer requests are needed to retrieve management information from the remote node.

If the *node* is an SNMPv1-only agent, then this command automatically downgrades the GetBulk request to an SNMPv1-supported GetNext request.

*node* can be an IP addressable system that supports SNMP. IP nodes may be identified by either Internet address or hostname.

One or more *object-ids* may be supplied as arguments to the command. Each *object-id* is an object identifier in dotted decimal format or by mnemonic name. If specified by mnemonic name, the MIB that defines the object identifier must be loaded using the MIB loader.

This command can only be run by users who belong to System, Administrator or Web Service Client roles. Users who are in Level1, Level2 or Guest roles will not be able to run this command.

## Parameters

-d

Dumps all SNMP packets to standard output in a hexadecimal and decoded ASN.1 format.

-v *version*

Requests that a particular version of SNMP be used to communicate with the remote node. Valid choices for version are 1, 2, 2c or 3.

For nodes not in the topology, the default will be 2c if not specified

-c *community*

Specifies the community string to use for authentication on the remote node.

*-port port*

Specifies the port to use in communication with the remote node.

*-t timeout*

Specifies a timeout period, in milliseconds, for communication with the remote node.

*-r retries*

Specifies the number of retries to use for communication with the remote node.

*-T*

Prints the OID in dotted decimal format.

*-n*

Nonrepeaters specifies the number of variables for which a single lexicographic successor is to be returned.

*-m*

Max-Repetitions specifies the number of lexicographic successors to be returned for the remaining variables.

*-pp Proxy Port*

Specifies the Proxy Port to use in communication with the node

*-pa Proxy Address*

Specifies the Proxy IP Address to use in communication with the node

*-a Authentication Protocol*

SNMPv3 Authentication Protocol (MD5|SHA)

*-A Authentication Passphrase*

SNMPv3 Authentication Passphrase

*-x Privacy Protocol*

SNMPv3 Privacy Protocol (DES|3DES|AES128|AES192|AES256)

*-X Privacy Passphrase*

SNMPv3 Privacy Passphrase

*-N context*

SNMPv3 Context Name (e.g. vlan1)

*-v3u SNMPv3 user name*

SNMPv3 security name (e.g. testV3user)

-u *<username>*

Supply the NNMi username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

-p *<password>*

Supply the NNMi user password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`nnmsnmpbulk.ovpl` employs an exponential backoff algorithm, based on *timeout* and *retries* to resend the SNMP request if a response is not received. For example, if *timeout* is 2000 (two seconds) and *retries* is 3, the initial request would time out after two seconds, the first retry would time out after four seconds, the second retry would time out after eight seconds, and the last retry would time out after sixteen seconds. Additional time will also be required for resolving the configuration.

## EXAMPLES

The following command retrieves all information under the system MIB object identifier from node `testnode`:

```
nnmsnmpbulk.ovpl -c community testnode .1.3.6.1.2.1.1.0
```

## AUTHOR

`nnmsnmpbulk.ovpl` was developed by Hewlett-Packard Company.

## FILES

The environment variable below represents a universal pathname that is established according to your shell and platform requirements:

```
$NNM_BIN/nnmsnmpbulk.ovpl
```

## SEE ALSO

[nnmsnmpwalk.ovpl\(1M\)](#), [nnmsnmpset.ovpl\(1M\)](#), [nnmsnmpnotify.ovpl\(1M\)](#).

*RFC 1155, 1157, 1212: SNMP Version 1.*

*RFC 1901 - 1908, 2576, 2578, 3416 - 3418: SNMP Version 2.*

*RFC 3411 - 3415: SNMP Version 3.*

## EXTERNAL INFLUENCES

## Environmental Variables

`$LANG` determines the language in which messages appear. If `$LANG` is not specified or is set to an empty string, a default of `C` is used instead of `$LANG`. If any internationalization variable contains an invalid setting, `nmmsnmpbulk.ovpl` behaves as if all internationalization variables are set to `C`.

## International Code Set Support

Supports single-byte and multiple-byte character code sets.

NOTE: SNMP MIB values of the type `DISPLAY STRING` are restricted to NVT-ASCII.

[Return to Reference Pages Index](#)

## Name

`nnmsnmpwalk.ovpl` — query a node using multiple SNMP GETNEXT requests

## SYNOPSIS

```
nnmsnmpwalk.ovpl -u user_name -p passwd [options] node object-id
```

```
nnmsnmpget.ovpl -u user_name -p passwd [options] node object-id [,object-id]...
```

```
nnmsnmpnext.ovpl -u user_name -p passwd [options] node object-id [,object-id]...
```

*options*: [-d] [-v version] [-C community] [-port port(default:161)] [-t timeout(default:5000)] [-r retries(default:1)] [-T] [-pp Proxy Port] [-pa Proxy Address] [-a Authentication Protocol] [-A Authentication Pass phrase] [-X Privacy Protocol] [-X Privacy Passphrase] [-N Context Name] [-v3u SNMPv3 user name]

## DESCRIPTION

`nnmsnmpwalk.ovpl` issues repeated SNMP GETNEXT requests to retrieve values for all instances of MIB objects registered on the node `node`. `nnmsnmpwalk.ovpl` automatically determines, based on the value for the `-v` option and the type of remote node, whether to use SNMP Version 1 or Community-based SNMP Version 2 (SNMPv2c) or version 3. If you do not specify a variable, `nnmsnmpwalk.ovpl` retrieves all values under the `object.iso.org`. Otherwise, the value of the variable determines the starting point in the object identifier space that is searched. `nnmsnmpwalk.ovpl` terminates when all object information under the specified variable has been returned. For example, the entire system group can be retrieved by using `.1.3.6.1.2.1.1.1`.

`nnmsnmpget.ovpl` uses the SNMP Get request to query `node` for information.

Generally an SNMP instance number needs to be appended (such as using `.0` in `.1.3.6.1.2.1.1.1.0` to get the `system.sysDescr.0` value).

`nnmsnmpnext.ovpl` performs the same action as `nnmsnmpwalk.ovpl`, except `nnmsnmpnext.ovpl` returns only a single value.

`node` can be an IP-addressable system that supports SNMP, or a target name for which an SNMP proxy configuration is defined. You can identify IP nodes by Internet address or hostname.

You may supply one or more variables as arguments to the command. Each variable is an object identifier in dotted decimal format or mnemonic name. If specified by mnemonic name, the MIB that defines the object identifier must be loaded using the MIB loader.

If you attempt to search beyond the end of the remote node's MIB with `nnmsnmpwalk.ovpl` or `nnmsnmpnext.ovpl`, the message `End of MIB` is returned.

These commands can only be run by users who belong to System, Administrator or Web Service Client roles. Users who are in Level1, Level2 or Guest roles will not be able to run these commands.

# Options

- d  
Dumps all SNMP packets to standard output in a hexadecimal and decoded ASN.1 format.
- v *version*  
Requests that a particular version of SNMP be used to communicate with the remote node. Valid choices for version are 1, 2, 2c or 3.  
For nodes not in the topology, the default will be 2c if not specified
- c *community*  
Specifies the community string to use for authentication on the remote node.
- port *port*  
Specifies the port to use in communication with the remote node.
- t *timeout*  
Specifies a timeout period, in milliseconds, for communication with the remote node.
- r *retries*  
Specifies the number of retries to use for communication with the remote node.
- T  
Prints the OID in dotted decimal format.
- pp *Proxy Port*  
Specifies the Proxy Port to use in communication with the node
- pa *Proxy Address*  
Specifies the Proxy IP Address to use in communication with the node
- a *Authentication Protocol*  
SNMPv3 Authentication Protocol (MD5|SHA)
- A *Authentication Passphrase*  
SNMPv3 Authentication Passphrase
- x *Privacy Protocol*  
SNMPv3 Privacy Protocol (DES|3DES|AES128|AES192|AES256)
- X *Privacy Passphrase*

## SNMPv3 Privacy Passphrase

-N *context*

SNMPv3 Context Name (e.g. vlan1)

-v3u *SNMPv3 user name*

SNMPv3 security name (e.g. testV3user)

-u *<username>*

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

-p *<password>*

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`nnmsnmpwalk.ovpl` employs an exponential backoff algorithm, based on *timeout* and *retries* to resend the SNMP request if a response is not received. For example, if *timeout* is 2000 (two seconds) and *retries* is 3, the initial request would time out after two seconds, the first retry would time out after four seconds, the second retry would time out after eight seconds, and the last retry would time out after sixteen seconds. Additional time will also be required for resolving the configuration.

## EXAMPLES

The following command requests the system subtree for the node *testnode*:

```
nnmsnmpwalk.ovpl testnode system
```

Output for the above command typically resembles the following:

```
nnmsnmpwalk.ovpl -c community 10.97.1.7 .1.3.6.1.2.1.1
.1.3.6.1.2.1.1.1.0: ASN_OCTET_STR = Ethernet Switch 470-24T-PWR
.1.3.6.1.2.1.1.2.0: ASN_OBJECT_ID = .1.3.6.1.4.1.45.3.63.1
.1.3.6.1.2.1.1.3.0: ASN_TIMETICKS = 63050579
.1.3.6.1.2.1.1.4.0: ASN_OCTET_STR = Bob Jones 933-558-3453
.1.3.6.1.2.1.1.5.0: ASN_OCTET_STR = wr3-2-front-storage-n91-60-2
.1.3.6.1.2.1.1.6.0: ASN_OCTET_STR = Woods Run 3 2nd floor
.1.3.6.1.2.1.1.7.0: ASN_INTEGER = 3
```

## AUTHOR

`nnmsnmpwalk.ovpl`, `nnmsnmpget.ovpl`, and `nnmsnmpnext.ovpl` were developed by Hewlett-Packard Company.

## FILES

`$NNM_BIN/nnmsnmpwalk.ovpl`

\$NNM\_BIN/nmmsnmpnext.ovpl

For information about universal pathnames for your platform and shell, see the [nnm.envvars\(1\)](#) reference page.

## SEE ALSO

[nmmsnmpset.ovpl\(1M\)](#), [nmmsnmpbulk.ovpl\(1M\)](#), [nmmsnmpnotify.ovpl\(1M\)](#).

*RFC 1155, 1157, 1212: SNMP Version 1.*

*RFC 1901 - 1908, 2576, 2578, 3416 - 3418: SNMP Version 2.*

*RFC 3411 - 3415: SNMP Version 3.*

## EXTERNAL INFLUENCES

### Environmental Variables

\$LANG determines the language in which messages appear. If \$LANG is not specified or is set to an empty string, a default of c is used instead of \$LANG. If any internationalization variable contains an invalid setting, nmmsnmpget.ovpl behaves as if all internationalization variables are set to c.

### International Code Set Support

Supports single-byte and multiple-byte character code sets.

NOTE: SNMP MIB values of the type DISPLAY STRING are restricted to NVT-ASCII.

[Return to Reference Pages Index](#)

## Name

`nnmsnmpset.ovpl` — issue an SNMP set request

## SYNOPSIS

```
nnmsnmpset.ovpl -u user_name -p passwd [options] node object-id asnType value
```

```
options: [-d] [-v version] [-c write community] [-port port(default:161)] [-t  
timeout(default:5000)] [-r retries(default:1)] [-T] [-pp Proxy Port] [-pa Proxy Address] [-a  
Authentication Protocol] [-A Authentication Pass phrase] [-X Privacy Protocol] [-X Privacy  
Passphrase] [-N Context Name] [-v3u SNMPv3 user name]
```

## DESCRIPTION

`nnmsnmpset.ovpl` issues an SNMP set request to alter MIB objects on the remote *node*.

*node* can be an IP-addressable system that supports SNMP, or a target name for which an SNMP proxy configuration is defined. You can identify IP nodes by an Internet address or by a hostname.

You specify the data passed by `nnmsnmpset.ovpl` to the remote node as a triple of *object-id,asnType,value*. You must supply one or more triples as command-line arguments.

Each *object-id* is an object instance identifier in dotted decimal format (for example, `.1.3.6.1.4.1.11.2.17.2.1.0` or mnemonic string format (for example, `openViewSourceId.0`).

Each *asnType* must be one of the following *asnTypes*:

integer

integer32

unsigned32

octetstring

octetstringhex

octetstringoctal

octetstringascii

objectidentifier

null

ipaddress

counter

counter64 (for SNMPv2c or v3 capable remote nodes)

gauge

gauge32

timeticks

opaque

opaquehex

opaqueoctal

opaqueascii

For a complete description of each *asnType*, refer to *RFC 1155* and *RFC 1902*.

The *value* must be valid for the *asnType* specified. When using an *asnType* where a hexadecimal or octal value is needed, you must fully define each byte of the value. For example, `fff` (or `17377`) is not allowed, but `0fff` (or `017377`) is. For the *asnType* *null*, you must specify a *value* on the command line. This *value* is ignored when the request is created. *value* must not be larger than 512 bytes.

This command can only be run by users who belong to System, Administrator or Web Service Client roles. Users who are in Level1, Level2 or Guest roles will not be able to run this command.

## Parameters

`-d`

Dumps all SNMP packets to standard output in a hexadecimal and decoded ASN.1 format.

`-v version`

Requests that a particular version of SNMP be used to communicate with the remote node. Valid choices for version are `1`, `2`, `2c` or `3`.

For nodes not in the topology, the default will be `2c` if not specified

`-c write community`

Specifies the write community string to use for authentication on the remote node.

`-port port`

Specifies the port to use in communication with the remote node.

`-t timeout`

Specifies a timeout period, in milliseconds, for communication with the remote node.

`-r retries`

Specifies the number of retries to use for communication with the remote node.

-T

Prints the OID in dotted decimal format.

-pp *Proxy Port*

Specifies the Proxy Port to use in communication with the node

-pa *Proxy Address*

Specifies the Proxy IP Address to use in communication with the node

-a *Authentication Protocol*

SNMPv3 Authentication Protocol (MD5|SHA)

-A *Authentication Passphrase*

SNMPv3 Authentication Passphrase

-x *Privacy Protocol*

SNMPv3 Privacy Protocol (DES|3DES|AES128|AES192|AES256)

-X *Privacy Passphrase*

SNMPv3 Privacy Passphrase

-N *context*

SNMPv3 Context Name (e.g. vlan1)

-v3u *SNMPv3 user name*

SNMPv3 security name (e.g. testV3user)

-u *<username>*

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

-p *<password>*

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`nmmsnmpset.ovpl` employs an exponential backoff algorithm, based on *timeout* and *retries* to resend the SNMP request if a response is not received. For example, if *timeout* is 2000 (two seconds) and *retries* is 3, the initial request would time out after two seconds, the first retry would time out after four seconds, the second retry would time out after eight seconds, and the last retry would time out after sixteen seconds. Additional time will also be required for resolving the configuration.

## EXAMPLES

The following command sets the system contact to Bob Jones for the node testnode:

```
nnmsnmpset.ovpl c writeCommunity testnode system.sysContact.0 octetstring "Bob Jones"
```

The output for this command is as follows:

```
system.sysContact.0 OCTET STRING-(ascii): Bob Jones
```

## AUTHOR

nnmsnmpset.ovpl was developed by Hewlett-Packard Company.

## FILES

See the `nnm.envvars` reference page (and the UNIX manpage) for information about using environment variables for the following files:

`install_dir\bin\nnmsnmpset.ovpl`

`install_dir\doc\rfc*.txt`

[nnmsnmpwalk.ovpl\(1M\)](#), [nnmsnmpbulk.ovpl\(1M\)](#), [nnmsnmpnotify.ovpl\(1M\)](#).

*RFC 1155, 1157, 1212: SNMP Version 1.*

*RFC 1901 - 1908, 2576, 2578, 3416 - 3418: SNMP Version 2.*

*RFC 3411 - 3415: SNMP Version 3.*

## EXTERNAL INFLUENCES

### Environmental Variables

`$LANG` determines the language in which messages appear. If `$LANG` is not specified or is set to an empty string, a default of `c` is used instead of `$LANG`. If any internationalization variable contains an invalid setting, `nnmsnmpset.ovpl` behaves as if all internationalization variables are set to `c`.

### International Code Set Support

Supports single-byte and multiple-byte character code sets.

NOTE: SNMP MIB values of the `asnType octetstringascii` are restricted to NVT-ASCII.

[Return to Reference Pages Index](#)

## Name

`disco.NoVLANIndexing` — Specifies certain nodes where VLAN Indexing should be skipped during discovery polling.

## SYNOPSIS

```
disco.NoVLANIndexing
```

## DESCRIPTION

One of the methods NNMi uses to learn layer 2 connectivity between and among switch devices in a managed network is to retrieve the `dot1dTpFdbTable` (FDB) from the switches. However, for Cisco switches, NNMi must use a `VLAN-indexing` method to retrieve the entire FDB. Using this method, NNMi retrieves the FDB once for each configured VLAN on the Cisco device. If there is a large number of VLANs configured on each device, retrieving the FDB with `VLAN-indexing` might take a very long time, sometimes even hours, to complete.

Cisco switches are often configured to use the Cisco Discovery Protocol (CDP). CDP is considered to be a superior method for learning layer 2 connectivity. Large switches located in the in the core of the network might contain many VLANs. These switches typically do not have end nodes connected directly to them. If the switches you want to manage do not have end nodes connected directly to them, you might want to suppress the collection of the FDB on these large switches. NNMi still completes the layer 2 discovery using data collected from CDP. These large switches are prime candidates for suppression of `VLAN-indexing`. Do not suppress `VLAN-indexing` on smaller switches located at the network's edge (often known as access switches) that have many end nodes attached to them.

You can configure NNMi to suppress VLAN indexing. To do this, the NNMi administrator needs to create the `disco.NoVLANIndexing` file, where the name of the file is case-sensitive. The `ovjboss` service reads the `disco.NoVLANIndexing` file when it starts. If the NNMi administrator makes changes to the `disco.NoVLANIndexing` file after the `ovjboss` service starts, those changes will not take effect until the next time the `ovjboss` service starts. By default, the `disco.NoVLANIndexing` file does not exist. If the `disco.NoVLANIndexing` does not exist, this feature is disabled and NNMi attempts to use `VLAN-indexing` to collect the entire FDB table on all devices.

The `disco.NoVLANIndexing` file can contain IP addresses, IP address ranges, and comments. A comment consists of the pound (or hash) sign (#) and all characters between # and the end of the line. NNMi treats an empty line as a comment. IP addresses are specified in the standard IP version 4 dotted-decimal notation or standard IP version 6 format (RFC 2373).

For details on the format of IP address ranges, see the *Configure Address Ranges for Regions* section of the NNMi help.

NNMi considers a node to match if one of the listed IP addresses matches a node's management address. Other IP addresses hosted by the node are not considered. If a node matches one of the addresses in the `disco.NoVLANIndexing` file, NNMi collects only the default FDB (the FDB which is accessible by using the community string with no `@vlan-id` suffix appended).

Disabling the collecting of the entire FDB might cause some inaccuracies in the layer 2 layout of the

managed network. HP is not responsible for these inaccuracies. Carefully consider which switches you include in the `disco.NoVLANIndexing` file.

## EXAMPLES

The following is an example of a `disco.NoVLANIndexing` file:

```
#This entry suppresses VLAN-indexing for the node whose management address is
10.2.37.149
10.2.37.149

192.168.100-101.1 #This entry causes the nodes 192.168.100.1 and 192.168.101.1 to be
skipped, too

# Here are some examples of IPv6 addresses and ranges:
  2136::8:800:200C:417a
  fd01::a352:1245:fc4B
  2001:D88:2:0:a07:ffff:0a01:3200-37ff
```

## AUTHOR

`disco.NoVLANIndexing` was developed by Hewlett-Packard Company.

## FILES

`$NnmDataDir/shared/nnm/conf/disco/disco.NoVLANIndexing`

`%NnmDataDir%\shared\nnm\conf\disco\disco.NoVLANIndexing`

## SEE ALSO

See the *Maintaining NNMi* chapter in the newest version of the *NNMi Deployment Reference* for more information.

See the *Configure Address Ranges for Regions* section of the NNMi help.

[Return to Reference Pages Index](#)

## Name

`disco.SkipXdpProcessing` — Contains a list of management IP addresses for nodes NNMi should not query for discovery protocol information.

## SYNOPSIS

`disco.SkipXdpProcessing`

## DESCRIPTION

One method NNMi uses to discover layer 2 connectivity between and among network devices in a managed network is to collect information from the devices related to their discovery protocols. There are many defined discovery protocols. For example, Link Layer Discovery Protocol (LLDP) is an industry standard protocol, while there are many vendor-specific protocols like Cisco Discovery Protocol (CDP) for Cisco devices. These are all handled by NNMi discovery in the `XdpAnalyzer`.

You can configure NNMi to suppress discovery protocol collections for devices you specify. This feature makes use of a configuration file, `disco.SkipXdpProcessing`, that the NNMi administrator creates. The name of the file is case-sensitive. The `ovjboss` service reads the `disco.SkipXdpProcessing` when it starts up. If the NNMi administrator makes changes to this file after the `ovjboss` service starts up, those changes will not take effect until the next time the `ovjboss` service starts. By default, the `disco.SkipXdpProcessing` file does not exist. If the `disco.SkipXdpProcessing` does not exist, this feature is disabled and NNMi attempts to collect discovery protocol information from all managed nodes.

For more information about the known problems fixed by this feature, refer to the SEE ALSO section below.

The `disco.SkipXdpProcessing` file can contain IP addresses and comments. A comment consists of the pound (or hash) sign (#) and all characters between # and the end of the line. NNMi treats an empty line as a comment. Specify IP addresses in the standard IP version 4 dotted-decimal notation or standard IP version 6 format (RFC 2373).

NNMi considers a node to match if one of the listed IP addresses matches a node's management address. Other IP addresses hosted by the node are not considered. If a node matches one of the addresses in the `disco.SkipXdpProcessing` file, NNMi skips the `XdpAnalyzer` service for that node and does not collect discovery protocol information.

Disabling the discovery protocol processing of a node or nodes might cause some inaccuracies in the layer 2 layout of the managed network. HP is not responsible for these inaccuracies.

## EXAMPLES

The following is an example of a `disco.SkipXdpProcessing` file:

```
#This entry supresses the XdpAnalyzer processing for the node whose management address
is 10.2.37.149
10.2.37.149
```

```
192.168.100.1 #This entry causes the node 192.168.100.1 to be skipped, too
# Here are some examples of IPv6 addresses:
  2136::8:800:200C:417a
  fd01::a352:1245:fc4B
```

## AUTHOR

`disco.SkipXdpProcessing` was developed by Hewlett-Packard Company.

## FILES

```
$NnmDataDir/shared/nnm/conf/disco/disco.SkipXdpProcessing
%NnmDataDir%\shared\nnm\conf\disco\disco.SkipXdpProcessing
```

## SEE ALSO

See the *Maintaining NNMi* chapter in the newest version of the *NNMi Deployment Reference* for more information.

[Return to Reference Pages Index](#)

## Name

`hostnlookup.conf` — file containing hostnames or hostname wildcards that should not be resolved to IP addresses using the system IP name server

## SYNOPSIS

`hostnlookup.conf`

## DESCRIPTION

`hostnlookup.conf` is a file used by the `ovjboss` process to determine whether a hostname should be resolved to an IP address using the system IP name server. The NNMi process `ovjboss` attempts to match a hostname against each entry in the `hostnlookup.conf` file before attempting to resolve the hostname to an IP address. If a match is found, the process does not attempt to resolve the hostname to an IP address using the system IP name server.

Lines in the file contain one hostname or hostname wildcard per line. Each entry must be on a single line. Comments are denoted by a number sign (#) and cause the remainder of the line to be ignored. Blank lines are allowed.

Use the `hostnlookup.conf` file when you determine that a specific hostname (or set of hostnames) cannot be resolved to an IP address using the systems IP name server.

The administrator must create the `hostnlookup.conf` file. It does not exist by default.

If you modify the `hostnlookup.conf` file while `ovjboss` is running, you can use the `$NnmInstallDir/support/nmsdnssync.ovpl` command to load the updated file. This also reloads the `ipnlookup.conf` file.

## EXAMPLES

The following is an example of a `hostnlookup.conf` file:

```
# A single hostname
badsys.mydomain.mycorp.com
# An IP wildcard
*.baddomain.mycorp.com
```

In the first example, the system name is bad. This causes some DNS servers to respond with unexpected results. In the second example, there is a domain that cannot be resolved. If these names are found, do not allow the system to waste time attempting to resolve the hostnames.

## AUTHOR

`hostnlookup.conf` was developed by Hewlett-Packard Company.

## FILES

`$NnmDataDir/shared/nnm/conf/hostnolookup.conf`

## SEE ALSO

`ipnolookup.conf(4)`.

[Return to Reference Pages Index](#)

## Name

`ipnolookup.conf` — file containing IP addresses or IP wildcards that should not be resolved to hostnames using the system IP name server

## SYNOPSIS

`ipnolookup.conf`

## DESCRIPTION

`ipnolookup.conf` is a file used by all NNMi processes to determine whether an IP address should be resolved to a hostname using the system IP name server. NNMi processes attempt to match an IP address against each entry in the `ipnolookup.conf` file before attempting to resolve the IP address to a hostname. If a match is found, the process does not attempt to resolve the IP address to a hostname using the system IP name server.

Lines in the file contain one IP address or IP wildcard per line. Each entry must be on a single line. Comments are denoted by a number sign (#) and cause the remainder of the line to be ignored. Blank lines are allowed.

Use the `ipnolookup.conf` file when you determine that a specific IP address (or range of IP addresses) cannot be resolved to a hostname using the system IP name server.

The administrator must create the `ipnolookup.conf` file. It does not exist by default.

If you modify the `ipnolookup.conf` file while NNMi processes are running, you can run the `$NnmInstallDir/support/nmsdnssync.ovpl` command with no arguments to load the updates to the `ipnolookup.conf` file.

## EXAMPLES

The following is an example of a `ipnolookup.conf` file:

```
# A single IP address
192.168.1.100
# An IP wildcard
10.*.*.*
# An IP wildcard range
192.168.1.101-255
```

In the first example, the single IP address could be routed to the Internet because many web sites use `192.168.*.*` for IP address. In the second example, the IP wildcard range could be NAT addresses. As such, they are not suitable for communications. In the third example, the IP wildcard range could be a set of addresses used for some purpose other than the primary IP address.

## AUTHOR

ipnolookup.conf was developed by Hewlett-Packard Company.

## FILES

`$NnmDataDir/shared/nnm/conf/ipnolookup.conf`

## SEE ALSO

hostnolookup.conf(4).

[Return to Reference Pages Index](#)

## Name

`logging.properties` — file containing configurable logging properties used by the `NMSLogManager` service to control how NNMi uses the Java Util Logging framework to log to product log files

## SYNOPSIS

`logging.properties`

## DESCRIPTION

`logging.properties` is a file used by the `NMSLogManager ovjboss` service to configure how NNMi logs to product log files using Java Util Logging (JUL). After you make changes to this file, you must restart `ovjboss`, or run the `$NnmInstallDir/support/nnmrereadlogging.ovpl` command.

NOTE: This document does not explain the JUL framework. If you would like to learn more about this framework, go to <http://java.sun.com/j2se/6/docs/guide/logging/overview.html>

The format of the `logging.properties` file conforms to the standard format required by JUL, but this file also adds some custom extensions used by NNMi to support NNMi logging requirements. In addition to the basic JUL logging properties, NNMi adds the following `name=value` entries to the `logging.properties` file:

`<package.name>.<LogFileHandler>.count`

Specifies how many archive log files should be kept when `ovjboss` restarts. The default value is 2.

`<package.name>.<LogFileHandler>.limit`

Specifies how large (in bytes) the active log file should become prior to being rolled to a new active log file. The default value is 1048576 (1MB).

`<package.name>.<LogFileHandler>.pattern`

Specifies the pattern to use in naming log files for the `LogFileHandler`. The default value is `<logBaseName>.%g.%u.log`.

Where:

- `%g` corresponds to the current count of archived log sets (see count above).
- `%u` is controlled by `java.util.logging` based on file locking. The "u" in `%u` stands for "unique". This value is only incremented if the current file name is locked. When that happens, the process increments `%u` until it can get a file lock, so that a new log file can be created.

NOTE: While `%g` reflects a limit on the number of archived sets of logs, an increment of `%u` shows that the file is currently locked. In the `logging.properties` file, if you set 5 for count and 1MB for limit, you can get up to five sets of archive log files. Each set has as many 1MB log files as are required to store the logging data.

Like standard JUL properties files, you can also specify default log levels, parent logger relationships, log formatters, and other log handlers to use. If you are in doubt about changing any logging setting, **it's a good idea to check with support to make sure you make appropriate changes.**

## EXAMPLES

If you want to keep five sets of NNM log files and roll the active logs by size approximately every 5MB, edit the `logging.properties` file as follows:

```
com.hp.ov.nms.admin.log.NnmMainFileHandler.limit = 5000000
com.hp.ov.nms.admin.log.NnmMainFileHandler.count = 5
```

Assuming you have a very busy system and you restart it once a week, you could get a set of files that looks something like this:

```
nm.0.0.log Note: the current log file
nm.1.0.log Note: this may have been archived due to a process restart
completed this morning
nm.2.0.log Note: this may have been caused by hitting a file size limit of 5
MB
nm.3.0.log Note: this may have been archived due to a process restart
completed this morning
nm.4.0.log Note: this may be an archive of the very first log file, and may
have been created when the
first log file surpassed a file size limit of 5 MB
etc...
```

## AUTHOR

`logging.properties` was developed by Hewlett-Packard Company.

## FILES

`$NNM_DATA/shared/nm/conf/ovjboss/logging.properties`

NNMi specific Java Util Logging properties used by the `NMSLogManager` service.

## SEE ALSO

`ovjboss(1M)`.

[Return to Reference Pages Index](#)

## Name

lwssofmconf.xml — The lwssofmconf.xml file configures the Light-weight Single Sign-on functionality. LWSSO allows NNMi to sign onto other HP Software products or NNM iSPIs without requiring signing in again.

## SYNOPSIS

lwssofmconf.xml

## DESCRIPTION

To enable Single Sign-on between systems, the lwssofmconf.xml files on all systems must all have the same `initString` value. The same username must be configured on the remote system to which you would like to single sign on. See the Deployment Reference for more information.

You must restart ovjboss if this file is modified with `ovstop ovjboss` and `ovstart`

## AUTHOR

lwssofmconf.xml was developed by Hewlett-Packard Company.

## FILES

`%NnmDataDir%\shared\nnm\conf\lwssofmconf.xml` (Windows)

`/var/opt/OV/shared/nnm/conf/lwssofmconf.xml` (UNIX)

## WARNING

If you are browsing between multiple NNMi installations, you will either need to disable Single Sign-on by editing lwssofmconf.xml and setting `enableLWSSOFramework="false"`, or you will need to edit the file to ensure the `initString` parameter is the same across all systems (enabling LWSSO). Otherwise, browsing to a second NNMi installation will sign you out of the previous NNMi installation when you return to the first system.

If the domain name of the installation system is short domain such as "mycompany" without any dot (such as mymachine.home), the NNMi console will immediately sign the user out. The restrictions for Single Sign-on browser cookies require a domain name must contain at least one dot such as "mycompany.com". If your hostname was set as "myhost.mycompany", and your domain name is "mycompnay" without dot, you will need to modify `$NnmDataDir/shared/nnm/conf/lwssofmconf.xml`. In this XML file, please replace "mycompany" with the full "myhost.mycompany" for "domain" field (don't change `protectedDomains`) then restart ovjboss with `ovstop ovjboss;ovstart`.

## SEE ALSO

[nmmsetdomain.ovpl\(1M\)](#), [nmmofficialqdn.ovpl\(1M\)](#), [ovstatus\(1\)](#), [ovstart\(1M\)](#), [ovstop\(1M\)](#).

[Return to Reference Pages Index](#)

## Name

`nnm.ports` — The following information shows the ports the NNMi management server listens on. In the case of port conflicts, you can change many of these port numbers.

## SYNOPSIS

`nnm.ports`

## DESCRIPTION

To change these port numbers, follow these steps:

1. Edit the `%NnmDataDir%\conf\nnm\props\nms-local.properties` (Windows) or `/var/opt/OV/conf/nnm/props/nms-local.properties` (Unix) to modify ports used by NNMi.
2. Identify the line containing the port number you need to change.
3. If necessary, remove the `#!` characters at the beginning of the line.
4. Modify the port number; then save your changes.
5. From a command prompt, run `ovstop`, then `ovstart` to restart NNMi.

The following ports are currently defined:

`jboss.http.port=80`

This TCP port is used as the default HTTP port for Web UI and Web Services. You can change this port value by modifying the `nms-local.properties` file and during NNMi installation.

`jboss.https.port=443`

This TCP port is used as the default secure HTTPS port (SSL) for Web UI and Web Services. You can change this port value by modifying the `nms-local.properties` file.

`jboss.rmi.port=1098`

This is the default TCP port of the RMI naming service. You can change this port value by modifying the `nms-local.properties` file.

`jboss.jnp.port=1099`

This TCP port is the default listening port for the bootstrap JNP service (JNDI provider). You can change this port value by modifying the `nms-local.properties` file.

`jboss.ejb3.port=3873`

This TCP port is used for remote clients to communicate with the EJB 3 container, and is the default EJB3 remoting connector port. You can change this port value by modifying the `nms-`

`local.properties` file.

`jboss.jrmp.port=4444`

This TCP port is the RMI Object port (Java Remote Method Protocol) used by RMI to transfer data (JRMP invoker). You can change this port value by modifying the `nms-local.properties` file.

`jboss.pooled.port=4445`

This TCP port is the default port (invoker port) used when pooling RMI requests. You can change this port value by modifying the `nms-local.properties` file.

`jboss.socket.port=4446`

This TCP port is the default RMI remoting server connector port. The jboss Remoting service uses this port. You can change this port value by modifying the `nms-local.properties` file.

`jboss.bisocket.port=4457`

This TCP port is the default messaging bi-socket connector port used by the jboss Remoting service. You can change this port value by modifying the `nms-local.properties` file.

`jboss.jmsControl.port=4458`

This TCP port is the default JMS control part; it is used for Global Network Management communication. You can change this port value by modifying the `nms-local.properties` file.

`jboss.sslbisocket.port=4459`

This TCP port is the default messaging bi-socket connector; it is used for secure Global Network Management communication. You can change this port value by modifying the `nms-local.properties` file.

`jboss.ssljmsControl.port=4460`

This TCP port is the default JMS control port; it is used for used for secure Global Network Management communication. You can change this port value by modifying the `nms-local.properties` file.

`jboss.ws.port=8083`

This TCP port is the default jboss Web Service port. You can change this port value by modifying the `nms-local.properties` file.

## AUTHOR

`nms.ports` was developed by Hewlett-Packard Company.

## FILES

`%NmDataDir%\conf\nm\props\nms-local.properties (Windows)`

`/var/opt/OV/conf/nnm/props/nms-local.properties (UNIX)`

## SEE ALSO

[ovstatus\(1\)](#), [ovstart\(1M\)](#), [ovstop\(1M\)](#).

[Return to Reference Pages Index](#)

## Name

`nnm.properties` — file containing the username and password for command line tools

## SYNOPSIS

`nnm.properties`

## DESCRIPTION

If you run NNMi command line tools often, HP recommends that you create an `nnm.properties` file. This file contains a username and a password that you can use in place of the `-u` and `-p` command line options. Using this file, you can run many commands without entering your password. This file contains an encrypted version of the password and should be created by executing the [nnmsetcmduserpw.ovpl](#) script. This script creates or edits a file placed in an `.nnm` subdirectory under your home directory at the following locations:

- For Windows: `\drive:\Documents and Settings\username\.nnm\`
- For UNIX: `~/.nnm/`

The contents of this file must be defined as follows:

`nnm.username`

The account user name

`nnm.password`

The account encrypted password

Most command line tools require an administrative user with an `admin` role assigned to the tool.

## AUTHOR

`nnm.properties` was developed by Hewlett-Packard Company.

## FILES

`$HOME/.nnm/nnm.properties`

## SEE ALSO

[nnmsetcmduserpw.ovpl\(1\)](#).

[Return to Reference Pages Index](#)

## Name

`ldap.properties` — file containing the settings for communicating with and building LDAP queries to the directory service

## SYNOPSIS

`ldap.properties`

## DESCRIPTION

For detailed information about the contents of the `ldap.properties` file, see the *Integrating NNMi with a Directory Service through LDAP* chapter of the *NNMi Deployment Reference*.

After modifying this file, run the `nmldap.ovpl -reload` command to reload the configuration.

## AUTHOR

`ldap.properties` was developed by Hewlett-Packard Company.

## FILES

`$NnmDataDir/shared/nm/conf/ldap.properties`

## SEE ALSO

[nmldap.ovpl\(1M\)](#).

[Return to Reference Pages Index](#)

## Name

nnmtrapd.conf — Filter file to block traps based on IP address or OID

## SYNOPSIS

nnmtrapd.conf

## DESCRIPTION

nnmtrapd.conf file can be used to configure filters to block traps based on both IP address and trap OID.

The filters are entered one per line. Each filter consists of an IP address, range or wildcard followed by one or more comma separated list of trap OIDs or range of trap OIDs or wildcards.

The format of the filter is:

```
<IP Address, OID[,OID]*>
```

IP Address could be a single IP Address or a pattern in prefix/prefix-length notation or range-wildcard notation. The special notation of "\*" indicates all addresses. You may not combine the prefix/prefix-length notation with the range-wildcard notation in the same address. Host name cannot be specified instead of an address. Every filter entry should have a unique address (single, wild card or range). Examples of addresses in prefix/prefix-length notation are:

```
10.2.112.0/20
1080:0:a00::/44
```

The same addresses in range-wildcard notations are:

```
10.2.112-127.*
1080:0:a00-a0f:****:*
```

The trap OID can also be specified as a range or as wildcard. However, you may only use either a range or a wildcard in one OID. Only the last sub OID of an OID can be specified as a wildcard or range. The special notation ".\*" indicates all OIDs. Some examples are:

```
.1.3.6.1.4.1.11.2.17.1.0.58915834-58915868
.1.3.6.1.4.1.11.*
```

For generic traps like link up, you can append the enterprise OID of the vendor to the trap OID for blocking a specific vendor. Conversely, to block a generic trap from all vendors you need to append a wildcard to the trap OID.

Blocking all traps from all address is not allowed. So the following entry is ignored:

```
<*, .*>
```

## EXAMPLES

The following example blocks all generic traps from all subnets in the range 10.2.120 to 10.2.127:

```
<10.2.120.0/21, .1.3.6.1.6.3.1.1.5.*>
```

The following example blocks link up traps from all devices in the 10.6.112/21 subnet whose enterprise OID is .1.3.6.1.4.1.11.2.3.7.11.17:

```
<10.6.112.0/21, .1.3.6.1.6.3.1.1.5.4.1.3.6.1.4.1.11.2.3.7.11.17>
```

The following example blocks link up traps from all devices in the 10.6.112/21 subnet.

```
<10.6.112.0/21, .1.3.6.1.6.3.1.1.5.4.*>
```

The following example blocks all traps under the OID .1.3.6.1.4.1.11.2.17 and the authentication failure trap from a single IPv6 address.

```
<1080::8:800:200c:417a, .1.3.6.1.4.1.11.2.17.*, .1.3.6.1.6.3.1.1.5.5.*>
```

## AUTHOR

nnmtrapd.conf was developed by Hewlett-Packard Company.

## FILES

```
$NnmDataDir/shared/nm/conf/nnmtrapd.conf
```

## SEE ALSO

nnmtrapconfig.ovpl(1M).

[Return to Reference Pages Index](#)

## Name

trapFilter.conf — Filter file to block traps based on IP address or OID

## SYNOPSIS

trapFilter.conf

## DESCRIPTION

Use the `trapFilter.conf` file to configure filters to block traps based on both IP address and trap OID. This is similar to using the `nmtrapd.conf` file, except that traps blocked by the `trapFilter.conf` file do not get stored in the trap binary store, nor are they used to analyze trap rates. Trap rates are not affected by incoming traps blocked by the `trapFilter.conf` file. NNMi does not store incoming traps blocked by the `trapFilter.conf` file.

Enter the filters one filter per line. Each filter consists of an IP address, range, or wildcard followed by one or more comma separated trap OIDs, range of trap OIDs, or wildcards.

The format of the filter is as follows:

```
<IP Address, OID[,OID]*>
```

*IP Address* could be a single IP Address or a pattern in prefix/prefix-length notation or range-wildcard notation. The special notation of "\*" indicates all addresses. Do not combine the prefix/prefix-length notation with the range-wildcard notation in the same address. Do not specify the hostname instead of an address. Every filter entry should have a unique address (single, wildcard, or range). Examples of addresses in prefix/prefix-length notation are as follows:

```
10.2.112.0/20  
1080:0:a00::/44
```

The same addresses in range-wildcard notations are:

```
10.2.112-127.*  
1080:0:a00-a0f:****:*
```

Specify the trap OID as a range or as wildcard. Only use a range or a wildcard in one OID. Only specify the last sub OID of an OID as a wildcard or range. The special notation "."\* indicates all OIDs. Some examples are as follows:

```
.1.3.6.1.4.1.11.2.17.1.0.58915834-58915868  
.1.3.6.1.4.1.11.*
```

For generic traps like `linkUp`, you can append the enterprise OID of the vendor to the trap OID for blocking a specific vendor. Conversely, to block a generic trap from all vendors you need to append a wildcard to the trap OID.

Blocking all traps from all addresses is not allowed. So the following entry is ignored:

```
<*, .*>
```

## EXAMPLES

The following example blocks all generic traps from all subnets in the range 10.2.120 to 10.2.127:

```
<10.2.120.0/21, .1.3.6.1.6.3.1.1.5.*>
```

The following example blocks link up traps from all devices in the 10.6.112/21 subnet whose enterprise OID is .1.3.6.1.4.1.11.2.3.7.11.17:

```
<10.6.112.0/21, .1.3.6.1.6.3.1.1.5.4.1.3.6.1.4.1.11.2.3.7.11.17>
```

The following example blocks link up traps from all devices in the 10.6.112/21 subnet.

```
<10.6.112.0/21, .1.3.6.1.6.3.1.1.5.4.*>
```

The following example blocks all traps under the OID .1.3.6.1.4.1.11.2.17 and the authentication failure trap from a single IPv6 address.

```
<1080::8:800:200c:417a, .1.3.6.1.4.1.11.2.17.*, .1.3.6.1.6.3.1.1.5.5.*>
```

## AUTHOR

trapFilter.conf was developed by Hewlett-Packard Company.

## FILES

```
$NnmDataDir/shared/nnm/conf/trapFilter.conf
```

## SEE ALSO

nnmtrapconfig.ovpl(1M).

[Return to Reference Pages Index](#)

## Name

`UnnumberedNodeGroup.conf` — Node group name identifying those nodes having unnumbered interfaces that you want NNMi to discover.

## SYNOPSIS

`UnnumberedNodeGroup.conf`

## DESCRIPTION

NNMi uses the `UnnumberedNodeGroup.conf` file to determine whether devices should be evaluated for the existence of unnumbered interfaces. The evaluation results in NNMi showing the potential layer 2 connection across the nodes. This file must contain the name of a single node group containing the device identifiers, or it can be the name of a parent node group that represents multiple child node groups containing the device identifiers.

If this file is missing or the specified node group name does not exist in NNMi, NNMi will not enable the unnumbered interface management feature. In addition, the NNMi administrator must create the `UnnumberedNodeGroup.conf` file. It does not exist by default.

If you modify the `UnnumberedNodeGroup.conf` file while NNMi processes are running, you must restart NNMi.

## EXAMPLES

The following is an example of an `UnnumberedNodeGroup.conf` file:

```
# This is the name of an node group containing devices with unnumbered interfaces.  
Unnumbered Node Group
```

In the example, a node group named `Unnumbered Node Group` exists in NNMi. Add comment information as a separate line prefaced by a `#` character.

## AUTHOR

`UnnumberedNodeGroup.conf` was developed by Hewlett-Packard Company.

## FILES

`$NnmDataDir/shared/nm/conf/disco/UnnumberedNodeGroup.conf`

`%NnmDataDir%\shared\nm\conf\disco\UnnumberedNodeGroup.conf`

## **SEE ALSO**

UnnumberedSubnets.conf(4).

[Return to Reference Pages Index](#)

## Name

UnnumberedSubnets.conf — Contains a list of subnets that should be looked up by the NNMi against routing tables on the devices for unnumbered interface management

## SYNOPSIS

UnnumberedSubnets.conf

## DESCRIPTION

UnnumberedSubnets.conf is a file used by NNMi to limit the routing table query for unnumbered interface management. If you do not create and configure this file, NNMi will do a full MIB-II routing table walk against the discovered devices; By using the UnnumberedSubnets.conf file, NNMi requests MIB data for only those routes falling in the specified subnet ranges. It is a good practice to use this file and reduce the amount of discovery traffic and performance effect on the devices.

The UnnumberedSubnets.conf file can have one or more lines of CIDR subnets. The order of the subnets is not important, and NNMi will sort them out before querying the routing table. The range of subnet is inclusive.

The NNMi administrator must create the UnnumberedSubnets.conf file. It does not exist by default.

If you modify the UnnumberedSubnets.conf file while NNMi processes are running, you must restart NNMi.

## EXAMPLES

The following is an example of an UnnumberedSubnets.conf file:

```
#This entry filters the following routes: 10.1.0-63.  
10.1.5.0/18  
#This entry filters the following routes: 15.2.*.*  
15.2.126.0/16  
#This entry filters the following routes: 192.168.1.0-255  
192.168.1.0/24
```

In the above example, instead of a full routing table walk, NNMi only queries the ipRoutingTable or ipCidrRouting routing tables for routes in the specified ranges.

## AUTHOR

UnnumberedSubnets.conf was developed by Hewlett-Packard Company.

## FILES

```
$NnmDataDir/shared/nnm/conf/disco/UnnumberedSubnets.conf
```

```
%NnmDataDir%\shared\nnm\conf\disco\UnnumberedSubnets.conf
```

## SEE ALSO

UnnumberedNodeGroup.conf(4).

[Return to Reference Pages Index](#)

## Name

`nnmstatuspoll.ovpl` — update the status for a node using the State Poller

## SYNOPSIS

```
nnmstatuspoll.ovpl [ -node <nodename|IP Address> [-t timeout in secs] [-v] ]
```

## DESCRIPTION

The `nnmstatuspoll.ovpl` script enables you to dynamically poll a device that is being monitored. This results in a refresh of key collected state values. When all of the information for the state demand poll has been collected and displayed, the `nnmstatuspoll.ovpl` script informs you that the task that you requested is complete.

## Parameters

`-help`

Prints the usage statement.

`-t <timeout in secs>`

Client waits till given timeout in sec.

`-v`

Displays the detailed verbose log message on console.

`-jndiHost <hostname>`

Jboss server host. Default is localhost.

`-jndiPort <port>`

Jboss server port. Default is 1099.

`-node <nodename|IP Address>`

Target node name or IP address.

`-u <username>`

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-p <password>`

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

## RETURN VALUE

`nnmstatuspoll.ovpl` returns the appropriate output shown in the above Parameters section.

When using `-v` option, you see the information in the following columns:

Column 1: Indicates which protocol is used to collect the data.

Column 2: Indicates which device name was polled.

Column 3: Indicates which MIB instance was polled.

Column 4: Indicates the result of the poll.

Column 5: Indicates a mapped value, if it exists.

## AUTHOR

`nnmstatuspoll.ovpl` was developed by Hewlett-Packard Company.

## SEE ALSO

`nnm.properties(4)`

[Return to Reference Pages Index](#)

## Name

`nnmtopodump.ovpl` — Displays the contents of the NNMi topology database

## SYNOPSIS

```
nnmtopodump.ovpl -h | -u <username> -p <password> -type <type> [-legacy <format>] [-filter <filter>]
```

## DESCRIPTION

`nnmtopodump.ovpl` displays the contents of the topology database. By default, NNMi displays the output in xml format unless you specify the `-legacy` option.

## Parameters

The `nnmtopodump.ovpl` script supports the following options:

- `-h`  
Displays the usage statement.
- `-u <username>`  
Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.
- `-p <password>`  
Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.
- `-http.host <host>`  
The server host; default is localhost.
- `-http.port <port>`  
The server host; default is 8004.
- `-type <type>`  
The type of the objects available to be printed. Use one of the following types:  
node|interface|incident|ip|subnet|rrp|vlan|nodeSensor|interfaceAggregation|card|l2connection
- `-legacy [short|long]`  
If you use the legacy option, NNMi displays the data in text output. If you do not specify this option, then NNMi displays the output in xml form. Use the legacy option with the following type values only: node, interface, ip, l2connection and interfaceAggregation. For `-legacy short`, this is only valid

for the type values node, interface, and l2connection.

`-filter <filter>`

Filters the output by property. The `nnmtopodump.ovpl` script supports the following filters:

node - node.name | node.shortname | node.id | node.uuid | node.status

interface - node.name | node.shortname | node.id | interface.ifType | interface.id | interface.uuid

ip - interface.id | node.id | ip.value | ip.id

vlan - node.name | node.id | vlan.id | vlan.name | vlan.value

nodeSensor - node.name | node.hostname | node.id | nodeSensor.id | nodeSensor.name | nodeSensor.type

card - node.name | node.hostname | node.id | card.id | card.name

l2connection - l2connection.name | l2connection.id | l2connection.uuid

interfaceAggregation - master.id | master.uuid | master.index | master.alias | slave.id | slave.uuid | slave.index | slave.alias

## EXAMPLES

```
nnmtopodump.ovpl -u username -p password -type node
```

Displays all of the nodes in the topology database in xml format. (You must provide an NNMi username and password.)

```
nnmtopodump.ovpl -u username -p password -legacy long -type node
```

To display the nodes in text format, you must use the legacy option. When you use the legacy option with `-type node`, NNMi displays the nodes and their interfaces.

Equivalent command in NNM 6.x/7.x: `ovtopodump -l`

```
nnmtopodump.ovpl -u username -p password -type node -filter node.name=foo.hp.com
```

Display information about node `foo.hp.com` in xml format.

```
nnmtopodump.ovpl -u username -p password -legacy long -type node -filter node.name=foo.hp.com
```

Display information about node `foo.hp.com` in text format. When you use the legacy option with `-type node`, it displays the nodes with the interfaces attached to the node.

Equivalent command in NNM 6.x/7.x: `ovtopodump -lr foo.hp.com`

```
nnmtopodump.ovpl -u username -p password -legacy long -type node -filter node.id=2345
```

Display information about the node having `nodeid` as 2345 in text format. NNMi also displays all interfaces of the node.

Equivalent command in NNM 6.x/7.x: `ovtopodump -lr 2345`

## **AUTHOR**

`nmmtopodump.ovpl` was developed by Hewlett-Packard Company.

## **SEE ALSO**

[Return to Reference Pages Index](#)

## Name

nnmtrapconfig.ovpl — Configure HP NNM Trap service

## SYNOPSIS

```
nnmtrapconfig.ovpl -u <user> -p <password> [-showProp] [-start] [-stop] [-readFilter] [-dumpBlockList]
[-resetBlockCache]
```

```
nnmtrapconfig.ovpl -setProp -u <user> -p <password> [-trapInterface <ip_addr>] [-unsetTrapInterface]
[-trapPort <port>] [-recvSocketBufSize <size>] [-blockTraps] [-unblockTraps] [-thresholdRate <rate>] [-
rearmRate <rate>] [-overallThresholdRate <rate>] [-overallRearmRate <rate>] [-windowSize <time>] [-
updateSourcesPeriod <time>] [-notifySourcesPeriod <time>] [-minTrapCount <count>] [-numSources
<count>] [-databaseQSize <count>] [-pipelineQSize <count>] [-databaseFileSize <size>] [-
databaseFileCount <count>] [-persist] [-loopbackAddrOverride <ip_addr>] [-resetLoopbackAddrOverride]
[-discoHintCacheSize <size>] [-discoHintCacheTimeout <time>]
```

## DESCRIPTION

nnmtrapconfig.ovpl can be used to display or modify the current properties of the Trap Service. In addition it can be used to start or stop the Trap Service. It also provides the following filter related functionalities: read filter configuration files, print out the current filter configuration and blocking caches or reset the blocking caches

## Parameters

nnmtrapconfig.ovpl supports the following parameters:

-u <username>

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

-p <password>

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

-showProp

Show the properties associated with Trap Service and their current values.

-start

Start the Trap Service.

-stop

Stop the Trap Service.

`-readFilter`

Cause Trap Service to read the filter configuration files.

`-dumpBlockList`

Print out the filter configuration and the blocking caches that the Trap Service creates.

`-resetBlockCache`

Reset the blocking caches in the Trap Service.

To set values for one or more properties, run `nmtrapconfig.ovpl` with the following options

`-u user`

The name of a user with system or administrator roles.

`-p password`

The password for the specified user.

`-setProp`

Set values for properties associated with Trap Service.

`-trapInterface ip address`

Set IP address on which the Trap Service listens for traps. By default it listens on all interfaces.

`-unsetTrapInterface`

Set Trap Service to listen on all interfaces.

`-trapPort port`

Set the port on which the Trap Service listens for traps. By default this is 162.

`-recvSocketBufSize size`

Set the size of socket buffer, in kilobytes, on which Trap Service listens for traps.

`-blockTraps`

Block traps based on filter and threshold configurations.

`-unblockTraps`

Do not block traps.

`-thresholdRate rate`

Set the rate in traps/sec at which trap sources or trap oids are blocked. This feature is only available with the NNM iSPI NET license.

`-rearmRate rate`

Set the rate in traps/sec at which blocked trap sources or trap oids are unblocked. This rate should be

less than or equal to the `thresholdRate`. This feature is only available with the NNM iSPI NET license.

`-overallThresholdRate rate`

Set the rate in traps/sec at which all incoming traps are blocked.

`-overallRearmRate rate`

Set the rate in traps/sec at which all incoming traps are unblocked. This rate should be less than or equal to the `overallThresholdRate`

`-windowSize time`

Set monitoring window size in seconds. This determines the window size in which trap sources are monitored. This means that whenever this time expires the counters keeping track of incoming traps are reset and a new window is started.

`-updateSourcesPeriod time`

Set the time period in seconds after which the list of blocked traps and sources are updated. This feature is only available with the NNM iSPI NET license.

`-notifySourcesPeriod time`

Set the time period in seconds after which the list of blocked traps and sources are reported in the `trapanalytics` log file under the `NNM_LOG` directory. This reporting is only available with the NNM iSPI NET license. This is also the time period after which a trap storm incident is generated, in case a trap storm occurs and the `overallThresholdRate` is violated.

`-minTrapCount count`

Set the minimum number of traps to be received from a source before it is considered for blocking. Also the minimum number of the same trap received before that trap is considered for blocking.

`-numSources count`

Set the number of noisiest sources to monitor at any given time. These will be the sources that are sending the most traps as well as the traps that occur most frequently. This information is reported in the `trapanalytics` log file in the `NNM_LOG` directory. This feature is only available with the NNM iSPI NET license.

`-databaseQSize count`

Set the maximum number of traps that can be held in the queue that writes traps to the database.

`-pipelineQSize count`

Set the maximum number of traps that can be held in the queues for each stage of the trap pipeline.

`-databaseFileSize size`

Set the maximum size, in MB, for one file in the trap database. When the file size reaches this value a rollover happens.

`-databaseFileCount count`

Set the maximum number of files in the trap database.

`-persist`

Persist the current properties so that on future restarts these values will be used.

`-loopbackAddrOverride ip address`

Sets the IP Address to replace the source address of a trap if the source address is a loopback address before forwarding the trap.

`-resetLoopbackAddrOverride`

Resets the loopback override address. When this options is executed, the user supplied loopback override address will be removed. In this case, NNM chooses one of the server's addresses as the loopback override address.

`-discoHintCacheSize size`

Sets the size of the cache used to store source hints for discovery. By default the cache has 5000 entries.

`-discoHintCacheTimeout time`

Sets the timeout value, in seconds, for entries in the cache used to store source hints for discovery. By default the timeout value is 3600 seconds.

## EXAMPLES

Show the properties associated with the Trap Service and their values:

```
nmtrapconfig.ovpl -u user -p pass -showProp
```

Start the Trap Service:

```
nmtrapconfig.ovpl -u user -p pass -start
```

Set the trap port to 1162:

```
nmtrapconfig.ovpl -u user -p pass -setProp -trapPort 1162
```

Enable blocking and also persist the current values for future invocations of Trap Service

```
nmtrapconfig.ovpl -u user -p pass -setProp -blockTraps -persist
```

Persist the current values for future invocations of Trap Service

```
nmtrapconfig.ovpl -u user -p pass -setProp -persist
```

## FILES

The following file stores properties for NNM Trap Service:

Windows: `data_dir\shared\nnm\conf\nnmtrapserver.properties`

UNIX: `/var/opt/OV/shared/nnm/conf/nnmtrapserver.properties`

The blocking filters can be configured in the following file:

Windows: `data_dir\shared\nnm\conf\nnmtrapd.conf`

UNIX: `/var/opt/OV/shared/nnm/conf/nnmtrapd.conf`

## AUTHOR

`nmtrapconfig.ovpl` was developed by Hewlett-Packard Company.

[Return to Reference Pages Index](#)

## Name

`nnmtrapload.ovpl` — Merge traps/events from a `trapd.conf` file with the current database

## SYNOPSIS

```
nnmtrapload.ovpl [ [ [ -loadTrapd <trapd_file> [ -authorLabel <author_label> -authorKey  
<author_key> ] ] -verbose <true/false> -skipExisting <true/false> -disableAllTraps <true/false> [-u  
<user>] [-p <password>] ] ]
```

## DESCRIPTION

Parses and merges events/traps from an existing `trapd.conf` file into the database.

`nnmtrapload.ovpl` should only be used for importing snmp trapd configurations, and NOT 6.x/7.x legacy ov event trap definitions.

If the provided category is not supported it is by default mapped to Status

If the provided severity is not supported it is by default mapped to Normal

The trap/event parser only resolves the tokens EVENT, FORMAT, SDESC, and EDESC, it does not resolve the tokens; EXEC, FORWARD, DISPLAY, or NODES

## Parameters

`nnmtrapload.ovpl` supports the following parameters:

`-u <username>`

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-p <password>`

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-loadTrapd trapd_file`

File location for `trapd.conf`

`-authorLabel author_label`

Author label for storing incident configs in the database.

`-authorKey author_key`

Author key for storing incident configs in the database.

`-verbose true/false`

If true logs each trap/event with verbose output; default is false

`-skipExisting true/false`

If true skips over traps/events that already exist in the database (updates/merges are not performed); default is false

`-disableAllTraps true/false`

If specified all traps should be loaded as disabled; default is false

## EXAMPLES

To Load trapd.conf file with verbose output

```
nmtrapdload.ovpl -loadTrapd /tmp/conf/trapd.conf -verbose true -u user -p password
```

To load trapd.conf file while skipping existing entries

```
nmtrapdload.ovpl -loadTrapd /tmp/conf/trapd.conf -skipExisting true -u user -p password
```

To load trapd.conf file with all events disabled

```
nmtrapdload.ovpl -loadTrapd /tmp/conf/trapd.conf -disableAllTraps true -u user -p password
```

To load trapd.conf file with a supplied author

```
nmtrapdload.ovpl -loadTrapd /tmp/conf/trapd.conf -authorLabel trapd_parser -authorKey com.hp.ov.nnm.parser -u user -p password
```

## AUTHOR

nmtrapdload.ovpl was developed by Hewlett-Packard Company.

[Return to Reference Pages Index](#)

## Name

nnmtrimincidents.ovpl — delete and archive incidents

## SYNOPSIS

```
nnmtrimincidents.ovpl [ [ [-age age -incr incr] | -date date ] [-nature nature] [-lifecycle  
lifecycleState] [-origin origin] [-name name] [-family family] [-sysobjectid sysobjectid] [-path  
path] [-archiveOnly] [-trimOnly] [-batch batchSize] [-u username] [-p password] [-quiet] ]
```

## DESCRIPTION

nnmtrimincidents.ovpl is used to delete incidents from the incident table. The deleted incidents are saved in a compressed archive file:

```
data_dir\tmp\incidentArchive.txt.gz
```

## Parameters

-age *age*

Specifies the age of incidents to trim. You should use this option in conjunction with the *incr* option. If specified, the value for *age* must be greater than 0.

-incr *increment*

Specifies the increment for *age* option. Supported increments include *days*, *weeks*, and *months*.

-date *date*

Specifies the date from which older incidents are trimmed. The date is specified in ISO 8601 standard format: *yyyy-mm-ddThh:mm:ss[+ or -]hh:mm*.

-archiveOnly

Creates an archive file. Does not trim incidents. This option requires you to specify the *age* or *date* option.

-trimOnly

Trims incidents without archiving the deleted incidents.

-batch *batchSize*

Specifies the batch size when trimming incidents. If specified, the value for *batch* must be greater than 0 and less than or equal to 1000.

-path *path*

Specifies the archive file name with a complete path. This path overrides the default archive file:

```
data_dir\tmp\incidentArchive.txt.gz
```

`-jndiHost hostname`

Server `jndi host`. Default is `localhost`.

`-jndiPort port`

Server `jndi port`. Default is `1099`.

`-lifecycle lifecycle`

Specifies the lifecycle state of incidents matching the `age|date` to trim.

Example `lifecycle` states include:

Registered

InProgress

Completed

Closed

`-name name`

Specifies the name of incidents matching the `age|date` to trim.

`-family family`

Optionally specify the family of incidents matching `age|date` to trim.

`-sysobjectid sysobjectid`

Optionally specify the device system object id of incidents matching `age|date` to trim.

`-nature nature`

Specifies the nature of incidents matching the `age|date` to trim.

Examples of `nature` include:

RootCause

SecondaryRootCause

Symptom

ServiceImpact

StreamCorrelation

None

Info

Dedup\_Stream\_Correlation

Rate\_Stream\_Correlation

`-origin origin`

Specifies the origin of incidents matching the `age | date` to trim.

Example `origin` states include:

ManagementSoftware

ManuallyCreated

RemotelyGenerated

SnmpTrap

Syslog

Other

`-u <username>`

Supply the NNMi administrator username required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-p <password>`

Supply the NNMi administrator password required to execute the command. Required unless a [nnm.properties\(4\)](#) file exists.

`-quiet quiet`

Specifies non-prompt mode.>

## EXAMPLES

Trim and archive incidents older than 6 days:

```
nnmtrimincidents.ovpl -age 6 -incr days
```

Trim and archive incidents older than 6 weeks with a nature of Symptom:

```
nnmtrimincidents.ovpl -age 6 -incr weeks -nature Symptom
```

Trim and archive incidents older than 6 months with a lifecycle state of Closed:

```
nnmtrimincidents.ovpl -age 6 -incr months -lifecycle Closed
```

Trim and archive incidents older than the specified date:

```
nnmtrimincidents.ovpl -date 2007-07-16T19:20:30
```

Trim and archive incidents older than the specified date with a nature of Symptom:

```
nnmtrimincidents.ovpl -date 2007-07-16T19:20:30+01:00 -nature Symptom
```

Trim and archive incidents older than the specified date with a lifecycle state of Closed:

```
nnmtrimincidents.ovpl -date 2007-07-16T19:20:30-01:00 -lifecycle Closed
```

Trim and archive incidents older than 6 days using the specified filename for the archive:

```
nmtrimincidents.ovpl -age 6 -incr days -path "C:\BkupDir\saveIncidents.gz"
```

Trim incidents older than 6 days (without archiving):

```
nmtrimincidents.ovpl -trimOnly -age 6 -incr days
```

Archive incidents older than 6 days (without trimming):

```
nmtrimincidents.ovpl -archiveOnly -age 6 -incr days
```

## AUTHOR

nmtrimincidents.ovpl was developed by Hewlett-Packard Company.

## SEE ALSO

[nmm.properties\(4\)](#)

[Return to Reference Pages Index](#)

## Name

`nnmwhat` — get specific NNM information from executables and libraries

## SYNOPSIS

`nnmwhat FILE`

## DESCRIPTION

The `nnmwhat` command is the equivalent command to the `what` command available on most Unix systems. This command is used to pull select string information from files such that specific information can be returned. This includes the copyright, operating system, patch level and product level. This is useful for the patching process and Support when determining the versions of files that are on the system.

## Parameters

Provide a file name to execute the command against. If no file is provided then STDIN is used.

## EXAMPLES

`nnmwhat ovaddobj.exe` will return the information from the `ovaddobj.exe` command.

[Return to Reference Pages Index](#)

## Name

ovaddobj — object registration utility

## SYNOPSIS

```
ovaddobj [ lrf-file ]
```

## DESCRIPTION

ovaddobj is used to register object managers (i.e. agents) with the HP process management process ovspmd(1M).

## Parameters

*lrf-file*

Specifies a Local Registration File (LRF), which must contain information about a single agent and the objects it manages.

### Note

You must specify all objects managed by the agent in the same LRF. Running `ovaddobj` against an LRF containing additional objects managed by a previously registered object manager does *not add* those objects. Instead, it *replaces* the previously registered objects with the new objects.

## EXAMPLES

```
ovaddobj mylrf
```

This registers the agent and all the objects described in the LRF *mylrf* into the NNM startup file.

## AUTHOR

ovaddobj was developed by Hewlett-Packard Company.

## FILES

See the `nmn.envvars` reference page (and the UNIX manpage) for information about using environment variables for the following files:

```
install_dir/bin/ovaddobj
```

## SEE ALSO

[ovstatus\(1\)](#), [ovstart\(1M\)](#), [ovstop\(1M\)](#), [ovdelobj\(1M\)](#), [ovspmd\(1M\)](#), [nmcluster\(1\)](#).

[Return to Reference Pages Index](#)

## Name

ovdelobj — object deregistration utility

## SYNOPSIS

```
ovdelobj [ lrf-file ]
```

## DESCRIPTION

ovdelobj is used to deregister the information for object managers (i.e. agents) from the HP process management process ovspmd(1M).

## Parameters

*lrf-file*

Specifies a Local Registration File (LRF), which contains information about a single agent and the objects it manages.

## EXAMPLES

```
ovdelobj mylrf
```

This deregisters the agent and all the objects described in the LRF `mylrf`.

## AUTHOR

ovdelobj was developed by Hewlett-Packard Company.

## FILES

See the `nmn.envvars` reference page (and the UNIX manpage) for information about using environment variables for the following file:

```
install_dir/bin/ovdelobj
```

## SEE ALSO

[ovstatus\(1\)](#), [ovstart\(1M\)](#), [ovstop\(1M\)](#), [ovaddobj\(1M\)](#), [ovspmd\(1M\)](#), [nmcluster\(1M\)](#).

[Return to Reference Pages Index](#)



## Name

ovjboss — wrapper around the jboss Application Server

## SYNOPSIS

ovjboss

## DESCRIPTION

ovjboss is a service component that falls under the management of `ovspmd`. It uses properties files (`$NNM_DATA/shared/nnm/conf/props/nms-support.properties`, `$NNM_DATA/shared/nnm/conf/props/nms-jboss.properties`, and `$NNM_DATA/shared/nnm/conf/props/ovjboss.jvmargs`) to pass arguments to the jboss application server. Each file contains documentation on how to change the settings it controls.

This command should be never be executed directly but it falls under the management of `ovspmd`. It will be started when running `ovstart` or `ovstart -c ovjboss`. To stop it either call `ovstop` or `ovstop -c ovjboss`. To see the status of internal services it monitors, call `ovstatus -v ovjboss`.

If there are problems starting `ovjboss`, one can look at the `ovjboss.log` and `jbosserver.log` log files and see if that might contain information to help solve the problem:

You must be logged on as root/administrator to run this command.

## EXAMPLES

To start `NNM` including `ovjboss` run the following command:

```
$InstallDir/bin/ovstart
```

To only start `ovjboss` run the following command:

```
$InstallDir/bin/ovstart -c ovjboss
```

To find the status of services started by `ovjboss` run the following command:

```
$InstallDir/bin/ovstatus -v ovjboss
```

## AUTHOR

ovjboss was developed by Hewlett-Packard Company.

## FILES

`$NNM_DATA/shared/nnm/conf/props/nms-jboss.properties`

Parameter file used by services started inside `ovjboss`.

`$NNM_DATA/shared/nm/conf/props/nms-support.properties`

Parameter file used by services started inside `ovjboss`.

`$NNM_DATA/shared/nm/conf/props/ovjboss.jvmargs`

Parameters passed to the JVM that jboss runs in

`$NNM_DATA/nm/conf/nms-local.properties`

Local configuration file, including Ports configuration

`$NNM_DATA/log/nm/jbossServer.log`

Log file containing exceptions (if any)

`$NNM_DATA/log/nm/ovjboss.log`

Log file containing stderr messages

## SEE ALSO

`ovspmd(1)`

`nms-local.properties(4)`

[Return to Reference Pages Index](#)

## Name

ovserror — Print out last errors from ovspmd process

## SYNOPSIS

ovserror

## DESCRIPTION

ovserror reports the last errors from the ovspmd process. It takes no parameters.

## RETURN VALUE

ovserror reports the last errors from the ovspmd process.

## AUTHOR

ovserror was developed by Hewlett-Packard Company.

## SEE ALSO

[ovstatus\(1\)](#), [ovstart\(1M\)](#), [ovstop\(1M\)](#).

[Return to Reference Pages Index](#)

# Name

ovspmd — NNM process management service

# SYNOPSIS

```
ovspmd [ [install] [start] [stop] [remove] [-W] [-d] [-V] [-f startup_file] ]
```

# DESCRIPTION

ovspmd manages the service processes that are part of NNM. It starts, stops, and reports status on these processes in response to requests from `ovstart`, `ovstop`, and `ovstatus`. `ovspmd` is normally started automatically by `ovstart`. On Windows, `ovspmd` is registered as a service. `ovspmd` registers under the service name HP OpenView Process Manager.

`ovstart` sends a request to `ovspmd` to start the object manager programs specified in the NNM startup file (SUF), by default `ovsuf`. NNM-managed processes are configured in a local registration file (LRF), and added to the SUF by `ovaddobj`. If you call `ovstart` with no arguments, `ovspmd` starts all managed processes configured to be started automatically (that is, with the initial start flag `OVs_YES_START` in the LRF).

`ovstop` sends a request to `ovspmd` to stop configured managed processes. If you call `ovstop` with no arguments, `ovspmd` stops all currently running managed processes, and then exits.

`ovstatus` sends a request to `ovspmd` to report the current running status of configured managed processes.

Managed processes are started by `ovspmd` as services (that is, in the background, with `stdin`, `stdout`, and `stderr` ignored).

Each managed process can be configured with a dependency list (that is, a list of other processes that must already be running before the process can be started successfully). `ovspmd` does not start a managed process until all the processes on which it depends have already initialized successfully. On startup, `ovspmd` verifies that no LRF-specified dependencies form a cycle. (An example of a cycle is A -> B -> C -> A.) These dependencies determine a relative sequencing for starting, as well as a reverse order for stopping.

`ovspmd` has a mechanism to automatically restart processes that fail unexpectedly. This process entails adding a retry count for the daemon processes as listed in the `$NNM_DATA/shared/nnm/conf/ovspmd.restart.properties` file. By default, the number of retries is 3. When a process dies unexpectedly, this count is decremented by one until it reaches zero. At that point, the process will not be automatically restarted. Attempting to start the process with `ovstart` will reset the retry count and start the process again. If the process has been running for two hours, then the process resets its retry counter. Removing entries will cause `ovspmd` not to do restarts. This is also true if the retry count is 0.

`ovspmd` distinguishes between three classes of object managers:

`OVs_WELL_BEHAVED`

A well-behaved process uses the `OVsPMD_API` (see `OVsPMD_API(3)`) to communicate with `ovspmd`. It sends `ovspmd` status information about successful and unsuccessful initialization, normal termination and abnormal termination, if configured to do so. `ovspmd` considers a well-behaved process to have

initialized successfully only when it explicitly reports that it has done so. A well-behaved process also exits when it receives the command `OVS_CMD_EXIT` from `ovspmd`.

The status information passed by the managed process to `ovspmd` is forwarded to `ovstart`, `ovstop`, or `ovstatus`, if currently running. The last message received from each managed process is saved, and then forwarded, on request, to `ovstatus`. The messages received from well-behaved processes are also logged to the application event log (which can be examined with the Event Viewer).

#### `OVS_NON_WELL_BEHAVED`

`ovspmd` can also manage object managers that do not use the `OVS_PMD` API (non-well-behaved processes) only if they do *not* go into the background of their own accord (see `OVS_DAEMON` below). Because a non-well-behaved process returns no status messages, `ovspmd` considers such a process to have initialized successfully if it is not exited within the LRF-specified timeout interval.

Non-well-behaved processes are terminated with Terminal Process if they do not exit within the configured timeout.

#### `OVS_DAEMON`

Managed processes that go into the background cannot be managed with a communication channel or with signals. `ovspmd` can start such a process, but it cannot stop or report meaningful status about the process because it does not have a communication channel or a process ID for it.

## Parameters

`install`

Install `ovspmd` as a service.

`start`

Start the `ovspmd` service.

`stop`

Stop the `ovspmd` service.

`remove`

Remove the `ovspmd` service.

`-W`

Do not start managed processes when `ovspmd` starts. Wait for `ovstart` to request it.

`-d`

Used for debugging. When used, `ovspmd` does *not* become a service.

`-V`

Run in very verbose mode. In this mode, `ovspmd` outputs very detailed information about the configuration of the managed processes. This is far too much information for ordinary use.

`-f startup_file`

Read *startup\_file* as the startup file (SUF) instead of the default. Note that *startup\_file* must be an absolute path.

## Application Authorization

`ovspmd` governs the management of NNM services. It uses the `ovspmd.auth` file to control which hosts, users, and applications can start and stop the NNM services. The `ovspmd.auth` file is located in `data_dir/conf\`.

`ovspmd` searches the entries in the `ovspmd.auth` file from beginning to end. As soon as it finds an entry that either explicitly allows or denies the access under consideration, it stops looking. Therefore, more specific entries should precede more general entries.

The file contains lines specifying the authorized hosts, users, and applications. Each line lists a single host, user, and application list authorized to connect to `ovspmd`. The format of each line of the file is:

```
#comment
```

```
hostname [username [appname1 appname2 appname3 ... ]]
```

The pound sign (#) and anything following it is a comment, which is ignored. Blank lines are also ignored.

`username` and `appname` are optional. If no application is present, the line permits (or denies) access by any application. If no username is present, the line permits (or denies) access by any user running any application.

If `hostname` is a plus sign (+), the line refers to access from any host. If `username` is a plus sign (+), the line refers to access by any user. If a hostname is preceded by a minus sign (-), the line explicitly denies all access from that host. (Any username or application names that also happen to appear on the line are ignored.) If a username is preceded by a minus sign (-), the line explicitly denies any access by that user from the specified host. (Any application names that also happen to appear on the line are ignored.)

If any applications are listed, the line permits access only to the applications listed (by the specified user from the specified host). Note that the application names listed in the authorization file must match the registered name of the application, except that white space in the registered application name must be replaced with underscores.

The `ovspmd.auth` file created at installation contains more examples of the file format, and some examples are also included in the **EXAMPLES** section.

## DIAGNOSTICS

`ovspmd` issues error messages about configuration errors and system call failures. These messages are intended to be self-explanatory. If it currently has an open communication channel with `ovstart`, `ovstop`, or `ovstatus`, `ovspmd` forwards these error messages through the communication channel to be output by the program.

`ovspmd` can process multiple requests (start, stop, or status) at a time. Additional requests are queued by type until the current request completes.

In addition, `ovspmd` logs processing, configuration, and system errors using `nettl` in the OVS subsystem at the `ERROR` level. Messages indicating normal events, such as successful initialization, are logged at the

INFORMATIVE level. Messages indicating initialization failure or abnormal termination are logged at the WARNING level.

## EXAMPLES

The following is an example of the contents of the `ovspmd.auth` file:

```
# Normally, you should authorize any application
# run by any user on the same host on which ovspmd is running.
# To do so, use a single line listing the
# name of the host on which this file is located
# (for example, "thishost"):
```

```
thishost
```

```
# Similarly, if you are running Management
# Consoles, you should authorize any application
# run by any user on all the client hosts and on
# the server host. For example, if your server
# system named "bigsystem" has one client named
# "hohum", list each of them on a separate line in
# this file on bigsystem:
```

```
bigsystem
hohum
```

```
# It is possible to permit specific users to run
# specific applications from a remote system. The
# following line permits the user "shem" from host
# "blimp" to run the applications "Toaster Manager"
# and "Blender". Note that, because the application's
# registered name "Toaster Manager" contains white
# space, you must replace the whitespace with the
# underscore character in the authorization file:
```

```
shem blimp Toaster_Manager Blender
```

```
# It is not possible to exclude specific applications,
# except by explicitly permitting all non-excluded
# applications.
```

```
# The following line denies access by the user "fred"
# from any host:
```

```
+ -fred
```

```
# The following line denies any application access
# from the host "badguy":
```

```
-badguy
```

## AUTHOR

ovspmd was developed by Hewlett-Packard Company.

## FILES

See the `nnm.envvars` reference page (and the UNIX manpage) for information about using environment variables for the following files:

```
install_dir\bin\ovspmd
```

```
install_dir\conf\ovsuf
```

See `$NNM_DATA/shared/nnm/conf/ovspmd.restart.properties` for restart property configuration.

## EXTERNAL INFLUENCES

### Environmental Variables

`$LANG` provides a default value if the internationalization variables, `LC_ALL`, `LC_CTYPE`, and `LC_MESSAGES` are unset, null, or invalid.

If `$LANG` is unset, null, or invalid, the default value of `C` (or `English_UnitedStates.1252` on Windows) is used.

`LC_ALL` (or `$LANG`) determines the locale of all other processes started by `ovspmd`.

`LC_CTYPE` determines the interpretation of text as single-byte characters, multiple-byte characters, or both; the classification of characters as printable; and the characters matched by character class expressions in regular expressions.

`LC_MESSAGES` determines the language in which messages are displayed.

All other environment variables are inherited from the shell executing `ovspmd` (or the initial `ovstart` that starts `ovspmd`). `ovspmd` and all service processes share this same environment. As a result, `ovspmd` must be stopped and restarted for any environment changes to take effect (see `ovstart(1M)`).

## SEE ALSO

[ovstatus\(1\)](#), [ovstart\(1M\)](#), [ovstop\(1M\)](#), [ovaddobj\(1M\)](#), [ovdelobj\(1M\)](#), [nnmcluster\(1M\)](#).

[Return to Reference Pages Index](#)

## Name

`ovstart` — start NNM managed processes

## SYNOPSIS

```
ovstart [ [-c] [-d] [-o ovspmd_path] [-v] [--][ovspmd_options...][managed_process_names...]
```

## DESCRIPTION

`ovstart` starts NNM managed processes. If called with one or more *managed\_process\_name* arguments, it starts the designated managed process after first starting any other managed processes on which it depends. If called with no arguments, it starts all the managed processes that are configured to start by default.

`ovstart` does not exit until all the managed processes it has tried to start have either responded or timed out (failed to respond within the LRF-specified timeout interval). By default, it produces no output unless a managed process fails. When you execute it from the command line, it is advisable to use the `-c` or `-v` option to track the progress of the operation. Running `ovstart` again after the successful completion of a previous attempt to `ovstart` is completely harmless.

`ovstart` sends a start request (`OVS_REQ_START`) to the process management service, `ovspmd`. If `ovspmd` is not already running, `ovstart` starts it first.

`ovstart` must be run by the administrator or super-user.

The managed processes are configured by `ovaddobj` from information in local registration files (see `lrf(4)`). A managed process is named by the first field in the LRF describing it.

If `ovstart` is used on a node configured for NNM clustering (see `nnmcluster(1M)`) then the behavior of `ovstart` is different than described above. Specifically, `ovstart` behaves exactly like the `"nnmcluster -daemon"` command.

In a NNM cluster environment `ovstart` returns immediately (after launching the NNM cluster in the background). Instead, the `nnmcluster` command will determine if/when to start the other NNM processes. Please monitor `ovstatus` output to determine if NNM processes have completed startup.

In a NNM clustered environment the other command-line options to `ovstart` are not supported.

Note that for fine-grain control of NNM cluster attributes use the `nnmcluster` command directly. The `ovstart` command in a NNM cluster environment is provided for convenience starting NNM using a familiar command.

## Parameters

`ovstart` recognizes the following options. Any unrecognized options are reported by a usage message.

`-c`

Produce one line of information about the success or failure of each managed process.

-d

Report the important stages in processing, including starting, contacting, and sending the start request to `ovspmd`, and closing the communication channel.

-o `ovspmd_path`

Specifies that the executable for `ovspmd` is in `ovspmd_path` instead of in the default location, `install_dir\bin`. If `ovspmd` is already running, this option is ignored.

-v

Produce several lines of information about the success or failure of each managed process.

- `ovspmd_options`

Any option not known by `ovstart` is passed to `ovspmd`. Since the `-d` option is valid for both programs, it will be interpreted as an `ovstart` option, and will *not* be passed on to `ovspmd`. Likewise, the `-v` option *will be* passed to `ovspmd` since it is not valid for `ovstart`. If an option is not recognized by either, a usage message will be printed from `ovspmd`, not `ovstart`.

--

Terminates the options section of the `ovstart` command line. Any arguments following the comment token (`--`) are interpreted as names of managed processes to start, and passed to `ovspmd`.

## RETURN VALUE

In a non NNM cluster environment `ovstart` exits with the status representing the number of object managers from the start list that were *not* started successfully. If all requested managed processes were started successfully, `ovstart` exits with the status 0 (zero).

In a NNM cluster environment `ovstart` always exit immediately with the status 0 (zero).

## DIAGNOSTICS

`ovstart` reports certain command-line errors (in particular, too many arguments) and system errors. The messages are prefixed with `ovstart:`, and are intended to be self-explanatory. `ovstart` also outputs error messages received from `ovspmd`. These messages are prefixed with `ovspmd:`. `ovstart` does not treat unrecognized options as errors, but `ovspmd` does.

Note that `ovspmd` can process multiple requests (`ovstart`, `ovstop`, or `ovstatus`) at a time. If any of these commands is being handled, the new request will be queued by type until the previous command has completed.

## EXAMPLES

`ovstart`

Request `ovspmd` to start all managed processes configured to start by default. If `ovspmd` is not already

running, start it with no options. Only failures are reported.

```
ovstart -v -V -- ovjboss
```

Request `ovspmd` to start the `ovjboss` process, which results in starting the Jboss application server and all of the NNM services that are deployed together within Jboss, after first starting any other managed processes that the `ovjboss` process depends on. If `ovspmd` is not already running, start it in verbose mode (`-v` option). Report program startup, whether successful or not (`-v` option). Note that the comment token (`--`) option is necessary so that `ovstart` does not interpret `ovjboss` as an argument to the unrecognized `-v` option.

## AUTHOR

`ovstart` was developed by Hewlett-Packard Company.

## FILES

See the `nnm.envvars` reference page (or the UNIX manpage) for information on using environment variables for the following files:

```
install_dir\bin\ovstart
```

```
install_dir\bin\ovspmd
```

## EXTERNAL INFLUENCES

### Environmental Variables

`$LANG` provides a default value if the internationalization variables, `LC_ALL`, `LC_CTYPE`, and `LC_MESSAGES` are unset, null, or invalid.

If `$LANG` is unset, null, or invalid, the default value of `C` (or `English_UnitedStates.1252` on Windows) is used.

`LC_ALL` (or `$LANG`) determines the locale of all other processes started by `ovspmd`.

`LC_CTYPE` determines the interpretation of text as single-byte and/or multi-byte characters, the classification of characters as printable, and the characters matched by character class expressions in regular expressions.

`LC_MESSAGES` determines the language in which messages are displayed.

If `ovstart` is executed, and no `ovspmd` process is currently running, `ovspmd` inherits the environment of the executing shell. All managed processes started by `ovspmd` inherit this same environment.

To change the environment for `ovspmd` or any managed process, you must restart `ovspmd` with the correct environment. This requires that all managed processes be stopped (`ovspmd` does not terminate until all managed processes have been shut down).

As a result, to change the environment for any managed process started from `ovstart/ovspmd`, you must do the following:

1. Execute `ovstop` with no arguments to shut down all managed processes and `ovspmd`.
2. Set up the correct environment variables.
3. Execute `ovstart` to restart `ovspmd` and any or all managed processes.

## NNM Cluster

If a `com.hp.ov.nms.cluster.name` is defined in the `$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties` file, then `ovstart` will defer startup to the `nmcluster` command.

## SEE ALSO

[ovstatus\(1\)](#), [ovstop\(1M\)](#), [ovaddobj\(1M\)](#), [ovdelobj\(1M\)](#), [ovspmd\(1M\)](#), [nmcluster\(1M\)](#).

[Return to Reference Pages Index](#)

## Name

ovstop — stop NNM managed processes

## SYNOPSIS

```
ovstop [ [-c] [-d] [-v] [managed_process_names...] [ [-nofailover|-failover|-cluster]]
```

## DESCRIPTION

ovstop stops the NNM managed processes. ovstop sends a stop request (`OVS_REQ_STOP`) to the process management process (UNIX operating systems) or service (Windows operating systems), `ovspmd`. If called with one or more *managed\_process\_name* arguments, it stops the designated managed processes after first stopping any dependent processes. If called with no arguments, or if one of the named arguments is `ovspmd`, it stops all managed processes currently running, including `ovspmd` itself.

When a managed process does not respond to the ovstop request within the LRF-specified timeout interval, `ovspmd` forces the process to terminate by sending it termination signals, first `SIGTERM`, then `SIGKILL` (see `kill(1)`). Note that ovstop reports forced termination only if the `-v` or `-c` options are used (for example, `ovstop -v [managed_process_name]`). Whenever a managed process times out during a stop request, it is advisable to increase its timeout value. To increase the number of seconds that `ovspmd` waits for a process to respond to an ovstop request, follow the instructions in `$NNM_LRF/ov*` (UNIX operating system) or `install_dir\lrf\ov*` (Windows operating systems).

Unlike `ovstart`, ovstop will *not* start `ovspmd` if it is not already running.

The managed processes are configured by `ovaddobj` from information in Local Registration Files (see `lrf(4)`). A managed process is named by the first field in the LRF describing it. Like `ovstart`, ovstop uses dependency information from the LRF. If other managed processes depend on a managed process that is stopped, `ovspmd` notes their dependency and terminates all appropriate managed processes in reverse LRF dependency order.

ovstop must be run by the Windows administrator or UNIX superuser.

If an `OVS_DAEMON` process is configured with a `Stop Command` in its LRF entry, ovstop runs the command (see `lrf(4)`). This feature is used to stop processes that are no longer in contact with `ovspmd`. The `Stop Command` is provided and configured by the developer of the process, if appropriate.

The names of the NNM managed processes that were started by previous `ovstart` operation can be obtained by running the `ovstatus -c` command.

The `ovstop ovjboss` command would stop the Jboss application server and all of the NNM services deployed together within Jboss. The names of Jboss deployed NNM services can be obtained by running the `ovstatus -v ovjboss` command. The NNM services could only be stopped altogether by running the `ovstop ovjboss` command. It is not supported to stop any of these NNM services individually, independent of the other NNM services.

If ovstop is used on a node configured for NNM clustering (see `nnmcluster(1M)`) then the behavior of ovstop is different than described above. Specifically, ovstop (with no parameters) behaves exactly like the

"`nmcluster -shutdown`" command.

In a NNM cluster environment `ovstop` returns immediately (after sending the NNM cluster a shutdown signal in the background). The `nmcluster` command then shuts down NNM processes which might trigger a failover of NNM services to the standby cluster node. Please monitor `ovstatus` output to determine if NNM processes have completed shutdown.

In a NNM clustered environment the only command-line options recognized by `ovstop` are `-nofailover`, `-failover`, and `-cluster`.

Note that for fine-grain control of NNM cluster attributes use the `nmcluster` command directly. The `ovstop` command in a NNM cluster environment is provided for convenience shutting down NNM services using a familiar command.

## Parameters

`ovstop` recognizes the options described below. The first argument that is not an option, and any succeeding arguments, are interpreted as names of managed processes to stop, and are passed to `ovspmd` in the stop request.

`-c`

Produce one line of information about the success or failure for each managed process.

`-d`

Report the important stages in its processing, including contacting and sending the stop request to `ovspmd`, and the closing the communication channel.

`-v`

Produce several lines of information about the success or failure of each managed process.

`-failover`

(NNM cluster only) Causes the local NNM node to shutdown NNM processes (if it is the active node) and the NNM cluster process will terminate. At the same time, automatic failover is enabled so that NNM services will transfer to the standby node.

`-nofailover`

(NNM cluster only) Causes the local NNM node to shutdown NNM processes (if it is the active node) and the NNM cluster process will terminate. At the same time, automatic failover is disabled so that NNM services will not transfer to the standby node.

`-cluster`

(NNM cluster only) Causes all nodes in the NNM cluster to shutdown. The NNM cluster process on the standby node(s) will be shutdown first, then the active node will stop NNM services, and finally the NNM cluster process on the active node will shutdown.

## RETURN VALUE

`ovstop` exits with a status representing the number of managed processes that were *not* stopped

successfully. If all requested managed processes were successfully stopped, `ovstop` exits with the status 0 (zero).

## DIAGNOSTICS

`ovstop` reports certain command-line errors (in particular, too many arguments) and system errors. The messages are prefixed with `ovstop:`, and are intended to be self-explanatory. `ovstop` also outputs error messages received from `ovspmd`. These messages are prefixed with `ovspmd:`. `ovstop` ignores unrecognized options.

If a managed process is in a `PAUSED`, `PAUSE_ERROR`, `PAUSE_TIMEOUT`, `RESUME_ERROR`, `RESUME_TIMEOUT`, or `DEPENDENCY_ERR` state, it is stopped. However, a warning message is printed to inform you that `ovstop` was used on a process that was not in a running state.

Note that `ovspmd` can process multiple requests (`ovstart`, `ovstop`, or `ovstatus`) at a time. If any of these commands is being handled, the new request will be queued by type until the previous command has completed.

## AUTHOR

`ovstop` was developed by Hewlett-Packard Company.

## FILES

The environment variables below represent universal pathnames that are established according to your shell and platform requirements. See the `nnm.envvars(1)` manpage for information on universal pathnames for your platform and shell.

See the `nnm.envvars` reference page ((or the UNIX manpage) for information about using environment variables for the following files:

Windows: `install_dir\bin\ovstop`

Windows: `install_dir\bin\ovspmd`

UNIX: `$NNM_BIN/ovstop`

UNIX: `$NNM_BIN/ovspmd`

## EXTERNAL INFLUENCES

### Environmental Variables

If a `com.hp.ov.nms.cluster.name` is defined in the `$NNMDataDir/shared/nnm/conf/props/nms-cluster.properties` file, then `ovstop` will defer startup to the `nnmcluster` command.

`$LANG` provides a default value if the internationalization variables, `LC_ALL`, `LC_CTYPE`, and `LC_MESSAGES` are unset, null, or invalid.

If `$LANG` is unset, null, or invalid, the default value of `C` (or `English_UnitedStates.1252` on Windows) is used.

`LC_ALL` (or `$LANG`) determines the locale of all other processes started by `ovspmd`.

`LC_CTYPE` determines the interpretation of text as single-byte characters, multiple-byte characters, or both; the classification of characters as printable; and the characters matched by character class expressions in regular expressions.

`LC_MESSAGES` determines the language in which messages are displayed.

## NNM Cluster

If a `NNMCLUSTER_NAME` is defined in the `ov.conf` file, then `ovstop` will defer startup to the `nmcluster` command.

## SEE ALSO

[ovstatus\(1\)](#), [ovstart\(1M\)](#), [ovaddobj\(1M\)](#), [ovdelobj\(1M\)](#), [ovspmd\(1M\)](#), [nmcluster\(1M\)](#).

[Return to Reference Pages Index](#)

## Name

`pmd` — NNM Postmaster service

`pmdmgr` — NNM Postmaster manager

## SYNOPSIS

```
pmd [ [-Lsize] [-Tsize] [-Sstack\;option[;option] ...]]
```

```
pmdmgr [ [-Lsize] [-Tsize] [-Sstack\;option[;option]...]]
```

## DESCRIPTION

`pmd` is the NNM Postmaster service that receives NNM events forwarded from remote NNM 6.X and 7.X management stations and forwards them to the Incident pipeline.

The NNM `OV_EVENT` stack, embedded within `pmd`, actually performs these services. In addition, `OV_EVENT` logs events to the event logs in `data_dir\shared\nnm\databases\eventdb`. Logged events can be viewed with `nmmdumpevents`.

`pmdmgr` is the Postmaster service manager. You can use `pmdmgr` to alter stack configurations of the running Postmaster. For example, after you start `pmd`, you can change a stack's trace and log mask with this command. You can alter only general stack options with this command. You can set options specific to a stack (for example, `OV_EVENT`) only at startup.

Only the superuser can use the `pmdmgr` command.

## Parameters

`pmd` and `pmdmgr` recognize the following options. Options that are specific to only `pmd` or `pmdmgr` are listed as such. You can use the other options for either command.

```
-Sstack;option[;option...]
```

Allows you to specify options that are particular to a stack. You can specify general and specific stack options for all stacks by using the `-s` option. For a list of all standardized stack options, see the section called *General Substack Options* of this reference page (and the UNIX manpage). A stack can also have options that are unique to itself.

```
-Qt
```

Allows you to truncate the trace files while `pmd` is running. This option is valid only for `pmdmgr`.

```
-Ql
```

Allows you to truncate the log files while `pmd` is running. This option is valid only for `pmdmgr`.

```
-Lsize
```

Sets the maximum size of the `pmd` log files, `pmd.log0` and `pmd.log1`. By default the size of each log file is 500K.

`-Tsize`

Sets the maximum size of the `pmd` trace files, `pmd.trc0` and `pmd.trc1`. By default the size of each trace file is 1000K.

## Supported Stacks

`pmd` can be composed of many stacks, where each stack provides a service for `pmd`. For NNM, the following stacks are included:

`OV_EVENT`

Receives NNM events forwarded from remote NNM 6.X and 7.X management stations and forwards them to the Incident pipeline.

`TRCLOG`

Provides the `pmd` tracing and logging functionality.

## General Substack Options

`pmd` (`pmdmgr`) recognizes the following options for all stacks. See the `-s` option (above) that allows you to specify stack options for each stack.

`E`

Enables a stack. It switches a stack ON. You cannot switch on stacks by using `pmdmgr` after starting `pmd`.

`D`

Disables a stack. It switches a stack OFF. You cannot switch stacks off by using `pmdmgr` after starting `pmd`.

`T mask`

Controls the trace mask for a particular stack. Each stack can have different kinds of tracing and logging enabled. To find out how to OR different types of bits into this mask, see the *Tracing and Logging* section of this reference page (and the UNIX manpage). By default, `WARNING`, `ERROR`, and `DISASTER` messages are logged. You can alter the trace and log mask value by using `pmdmgr` while `pmd` is running. The new trace and log mask takes effect immediately.

## Tracing and Logging

`pmd` creates trace and log files in `install_dir\log\`. These are `pmd.trc[0-1]` and `pmd.log[0-1]`. The trace files contain all trace and log information. The log files contain only the log information.

`pmd` traces and logs in a circular fashion, by using two files for wrapping. When the 0 file becomes full, the 0 file is moved to the 1 file, and the old 1 file is truncated and made the 0 file. As a result, you always have

the most current set of log or trace messages. The 0 file is always the current file.

By default, all stacks, as well as the `pmd` itself, log `WARNING`, `ERROR`, and `DISASTER` messages. By default, tracing is not enabled. The `pmd.trcX` files are not created and used until you enable tracing. The trace files are created on demand by turning on some tracing with the `-T` general stack option.

You can turn on additional tracing and logging by altering `pmd.lrf`. To alter a stack's tracing or logging, you need to change its trace mask. You can specify a trace mask for any stack in `pmd`. To create a mask that is a combination of the bits that are listed below, you add the bits together. In effect, you "OR" the bits. The result is then used as the argument to the substack `-T` option.

#### DISASTERS (0x1)

Log disasters. Disasters should not occur. If they do, contact HP and supply a tracing and logging file to improve the quality of the product.

#### ERRORS (0x2)

Log errors. These are errors local to `pmd`, and do not include errors that are defined by protocols. CMIP errors that result from a CMISE operation are not logged by setting this bit. Errors such as no route or aborts are logged with this bit.

#### WARNINGS (0x4)

Log warnings. These are unusual conditions the system administrator may need to know about. These are not necessarily error conditions.

#### INFORM (0x8)

Log informative messages. These can be messages of various kinds (for example, when signals are received by `pmd`, what the `argv/argc` vectors are that the `pmd` was started with, and so on).

#### STATE (0x10)

Log state changes. This logging applies mostly to associations and bindings to `pmd`.

#### HDRIN (0x20)

Trace header information for operations that flow from a stack to `pmd`.

#### HDROUT (0x40)

Trace header information for operations that flow from `pmd` to a stack.

#### PDUIN (0x80)

Trace PDU for operations that flow from a stack to `pmd`.

#### PDUOUT (0x100)

Trace PDU for operations that flow from `pmd` to a stack.

#### RQT (0x200)

Trace `pmdxxx_req` calls that are related to an RQT. These calls operate on RQTs.

**MEM (0x400)**

Trace memory allocation and de-allocation calls.

**CCES (0x800)**

Trace calls that are related to CCEs (for example, allocation, freeing, and finding).

**TIMERS (0x1000)**

Trace timer-related `pm_xxxx_timer` function calls.

**STACKCALLS (0x2000)**

Trace stack-supplied `SI_xxxx` functions as `pmd` calls them.

**OPERATION (0x4000)**

Trace `pm_stack_ind`, `pm_stack_cnf`, and `pm_failed_operation`.

**ALL\_KINDS (0xffffffff)**

Log and trace everything.

**OV\_EVENT Stack Tracing**

`OV_EVENT` supports the following trace values. (None are enabled by default.) To create a mask that is a combination of the bits that are listed below, you add the bits together which effectively OR's the bits. The result is then used as the argument to the substack T option, and traces are written to the `pmd` trace file.

**EVENTCONNECTIONS (0x00400000)**

Trace all application connections and disconnections to `OV_EVENT`.

**EVENTRECEIPT (0x00800000)**

Trace the receipt of each event by `OV_EVENT`.

**EVENTFLOW (0x01000000)**

Trace significant milestones as an event flows through and out `OV_EVENT`. (Includes `EVENTQUEUES` trace as defined below.)

**EVENTFILTERS (0x02000000)**

Trace events as they are filtered out; that is, when they are not forwarded to a destination application.

**EVENTQUEUES (0x08000000)**

Trace events as they are queued for a busy application and when they are removed from the queue the application is listening.

**OV\_EVENT Specific Stack Options**

The `OV_EVENT` stack supports the following options:

### *bsize*

Sets the total size of the `OV_EVENT` log files to the given size, in megabytes (default size is 16 MB). Minimum size is 1 megabyte. Once the size is exceeded, oldest events will be dropped from the event logs.

### *n*

Disables `OV_EVENT` logging to the event log files. The default is to log.

### *qnum*

Specifies the maximum number of events which can be queued for a connected application. When this number is exceeded, `OV_EVENT` will disconnect from the application. Events are queued when they cannot be sent to an application (receipt is blocked because the application is busy elsewhere). The default maximum is 4096 events. Valid values range from 0 to 65536.

## DIAGNOSTIC

For help with problems related to starting `pmd`, refer to the troubleshooting section of *HP OpenView Managing Your Network with NNM*.

## EXAMPLES

Switch on all possible (general and stack-specific) tracing for the `OV_EVENT` stack. You want to see all inbound and outbound PDUs to and from this stack, as well as all other information. Change the `pmd.lrf` file to the following:

```
pmd:pmd:
OVS_YES_START::-SOV_EVENT;T0xffffffff:OVS_WELL_BEHAVED::
```

For these changes to take effect, first stop the NNM platform using the `ovstop` command. Second, use the `ovdelobj` command to delete the old `pmd.lrf` entry. Third, use the `ovaddobj` command to add the modified `pmd.lrf` entry. Finally, use the `ovstart` command to restart the platform.

To change the `OV_EVENT` trace and log mask after `pmd` process is running, use the following command:

```
pmdmgr -SOV_EVENT\;T0xF
```

This command causes `INFORM` messages to be logged, as well as `DISASTER`, `WARNING`, and `ERROR` messages.

## AUTHOR

`pmd` was developed by Hewlett-Packard Company.

## FILES

See the `nnm.envvars` reference page for information on using environment variables for the following files:

`data_dir\shared\nnm\lrf\pmd.lrf`

`data_dir\shared\nnm\databases\eventdb`

`install_dir\log\pmd.log[01]`

`install_dir\log\pmd.trc[01]`

## SEE ALSO

[nnmdumpevents\(1M\)](#), [ovstatus\(1\)](#), [ovstart\(1M\)](#), [ovstop\(1M\)](#), [ovspmd\(1M\)](#).

[Return to Reference Pages Index](#)