Netcool/OMNIbus Probe for HP Operations Manager
4.0

Reference Guide
March 23, 2018



SC22-5497-05

Note

Before using this information and the product it supports, read the information in Appendix A, “Notices and Trademarks,” on page 21.

Edition notice

This edition (SC22-5497-05) applies to version 4.0 of IBM Tivoli Netcool/OMNIbus Probe for HP Operations Manager and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC22-5497-04.

© Copyright International Business Machines Corporation 2012, 2018.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this guide ..... **v**
Document control page v
Conventions used in this guide vi

Chapter 1. Probe for HP Operations Manager **1**
Summary 1
Installing probes 2
Uninstalling Menu Tools 3
Configuration tasks 3
Configuring ObjectServer acknowledged event triggers 3
Additional probe tools 4
Desktop tools 4
WebGUI tools 5
Limitations of the Acknowledge and Unacknowledge alert commands 7
Running the probe 7
Run as `root` 7
Data acquisition 7
Configuring HP Operations Manager 8
Encrypting the Opcpassword Property 8
Processing duplicate events and modify events received from MEI 8
Configuring the Retry and Inactivity properties 8
Peer-to-peer failover functionality 9
Managing the probe over an HTTP/HTTPS connection 9
Configuring the command interface 10
Messages in the log file 10
Storing commands in the nco_http properties file 10
Properties and command line options 11
Elements 13
Annotation Event Alarm Elements 13
Message Stream Alarm Elements 14
Custom Message Attribute Elements 16
Error messages 16
ProbeWatch messages 19

Appendix A. Notices and Trademarks **21**
Notices 21
Trademarks 22

About this guide

The following sections contain important information about using this guide.

Document control page

Use this information to track changes between versions of this guide.

The IBM Tivoli Netcool/OMNIbus Probe for HP Operations Manager documentation is provided in softcopy format only. To obtain the most recent version, visit the IBM® Tivoli® Knowledge Center:

http://www-01.ibm.com/support/knowledgecenter/SSSHTQ/omnibus/common/kc_welcome-444.html?lang=en

Table 1. Document modification history

Document version	Publication date	Comments
SC22-5497-00	November 30, 2012	First IBM publication.
SC22-5497-01	July 11, 2014	The following note was added for Linux users: This probe is only supported in a 64-bit environment. This means that you must be running a 64-bit version of the Linux Operating System and a 64-bit version of Netcool/OMNIbus (either V7.4 or V8.1).
SC22-5497-02	June 11, 2015	"Summary" on page 1 updated. "Processing duplicate events and modify events received from MEI" on page 8 added. "Custom Message Attribute Elements" on page 16 added. **Enhancements:** Version 3.0 of the Probe for HP Operations Manager includes enhancements for the following RFEs: • **65446**: Support for duplicate event notification added. When the probe receives a duplication notification event from the Message Event Interface (MEI), it uses the message ID to request the full messages that it sends to the ObjectServer for deduplication. • **66018**: Support for custom message attributes added. The probe extracts the custom attributes from the messages that it receives from HP Operations Manager and logs the custom attributes with their associated values.

© Copyright IBM Corp. 2012, 2018

Table 1. Document modification history (continued)		
Document version	**Publication date**	**Comments**
SC22-5497-03	July 28, 2016	Version updated to 4.0 in “Summary” on page 1. Support extended to version 9.2 of HP Operations Manager. “Uninstalling Menu Tools” on page 3 added. “Configuring ObjectServer acknowledged event triggers” on page 3 added. “Additional probe tools” on page 4 added. “Managing the probe over an HTTP/HTTPS connection” on page 9 added. Functionality to send details of modified events to the ObjectServer added. See “Processing duplicate events and modify events received from MEI” on page 8 Version 4.0 of the Probe for HP Operations Manager addresses the following Enhancement Requests: • **65447**: Enable the probe to update the HPOM system with the alarm status change made on the Event List of the Netcool/OMNIbus ObjectServer.
SC22-5497-04	August 18, 2017	Support for version 9.1 of HP Operations Manager added to “Summary” on page 1.
SC22-5497-05	March 23, 2018	The following topic about running the probe as root added: “Run as root” on page 7.

Conventions used in this guide

All probe guides use standard conventions for operating system-dependent environment variables and directory paths.

Operating system-dependent variables and paths

All probe guides use standard conventions for specifying environment variables and describing directory paths, depending on what operating systems the probe is supported on.

For probes supported on UNIX and Linux operating systems, probe guides use the standard UNIX conventions such as **$***variable* for environment variables and forward slashes (**/**) in directory paths. For example:

```
$OMNIHOME/probes
```

For probes supported only on Windows operating systems, probe guides use the standard Windows conventions such as **%***variable***%** for environment variables and backward slashes (****) in directory paths. For example:

```
%OMNIHOME%\probes
```

For probes supported on UNIX, Linux, and Windows operating systems, probe guides use the standard UNIX conventions for specifying environment variables and describing directory paths. When using the Windows command line with these probes, replace the UNIX conventions used in the guide with Windows conventions. If you are using the bash shell on a Windows system, you can use the UNIX conventions.

Note : The names of environment variables are not always the same in Windows and UNIX environments. For example, %TEMP% in Windows environments is equivalent to $TMPDIR in UNIX and Linux environments. Where such variables are described in the guide, both the UNIX and Windows conventions will be used.

Operating system-specific directory names

Where Tivoli Netcool/OMNIbus files are identified as located within an *arch* directory under NCHOME or OMNIHOME, *arch* is a variable that represents your operating system directory. For example:

`$OMNIHOME/probes/`*`arch`*

The following table lists the directory names used for each operating system.

Note : This probe may not support all of the operating systems specified in the table.

Table 2. Directory names for the arch variable

Operating system	**Directory name represented by *arch***
AIX® systems	`aix5`
Red Hat Linux® and SUSE systems	`linux2x86`
Linux for System z	`linux2s390`
Solaris systems	`solaris2`
Windows systems	`win32`

OMNIHOME location

Probes and older versions of Tivoli Netcool/OMNIbus use the OMNIHOME environment variable in many configuration files. Set the value of OMNIHOME as follows:

- On UNIX and Linux, set `$OMNIHOME` to `$NCHOME/omnibus`.
- On Windows, set `%OMNIHOME%` to `%NCHOME%\omnibus`.

Chapter 1. Probe for HP Operations Manager

HP Operations Manager is a software application that provides central operations and event management. HP Operations Manager consists of a central management server, which interacts with intelligent software agents installed on the managed nodes. The Probe for HP Operations Manager extracts XML data through a web service agent and allows for these events to be managed. The probe supports versions 8.0, 9.0, 9.1, and 9.2 of HP Operations Manager.

Note : For Linux users, this probe is only supported in a 64-bit environment. This means that you must be running a 64-bit version of the Linux Operating System and a 64-bit version of Netcool/OMNIbus (either V7.4 or V8.1).

This guide contains the following sections:

- "Summary" on page 1
- "Installing probes" on page 2
- "Data acquisition" on page 7
- "Properties and command line options" on page 11
- "Elements" on page 13
- "Error messages" on page 16
- "ProbeWatch messages" on page 19

Summary

Each probe works in a different way to acquire event data from its source, and therefore has specific features, default values, and changeable properties. Use this summary information to learn about this probe.

The following table provides a summary of the Probe for HP Operations Manager.

Table 3. Summary	
Probe target	HP Operations Manager version 8.0, 9.0, 9.1 and 9.2.
Probe executable name	`nco_p_hpom`
Package version	4.0
Probe supported on	For details of supported operating systems, see the following Release Notice on the IBM Software Support website: http://www-01.ibm.com/support/docview.wss?uid=swg21677136
Properties file	`$OMNIHOME/probes/`*`arch`*`/hpom.props`
Rules file	`$OMNIHOME/probes/`*`arch`*`/hpom.rules`

© Copyright IBM Corp. 2012, 2018

Table 3. Summary (continued)	
Additional configuration files	`UnixHpomTool.sql`: This file creates a UI menu from the desktop called HP Operation Manager that contains two menu items: **Acknowledge Alert** and **Unacknowledge Alert**. `NHttpHpomProbe.pl`: This script is called by the **Acknowledge Alert** and **Unacknowledge Alert** menu items. `hpom.bidir.rules`: This file is called from the `hpom.rules` file and enables the **Acknowledge** and **Unacknowledge** alert operations. `UnixHpomTools.xml`: This file registers the `UnixHpomTools.cgi` script tool in Web GUI. `UnixHpomTools.cgi`: This script is called by the **Acknowledge Alert** and **Unacknowledge Alert** menu items. It sends the command to event factory using the probe's URL. For details about these configuration files, see "Additional probe tools" on page 4.
Requirements	For details of any additional software that this probe requires, refer to the `description.txt` file that is supplied in its download package.
Connection method	API
Multicultural Support	Available
Peer-to-peer failover functionality	Available
IP environment	IPv4 and IPv6
Federal Information Processing Standards (FIPS)	IBM Tivoli Netcool/OMNIbus uses the FIPS 140-2 approved cryptographic provider: IBM Crypto for C (ICC) certificate 384 for cryptography. This certificate is listed on the NIST website at http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2004.htm. For details about configuring Netcool/OMNIbus for FIPS 140-2 mode, see the *IBM Tivoli Netcool/OMNIbus Installation and Deployment Guide*.

Installing probes

All probes are installed in a similar way. The process involves downloading the appropriate installation package for your operating system, installing the appropriate files for the version of Netcool/OMNIbus that you are running, and configuring the probe to suit your environment.

The installation process consists of the following steps:

1. Downloading the installation package for the probe from the Passport Advantage Online website.

   Each probe has a single installation package for each operating system supported. For details about how to locate and download the installation package for your operating system, visit the following page on the IBM Tivoli Knowledge Center:

   http://www-01.ibm.com/support/knowledgecenter/SSSHTQ/omnibus/probes/all_probes/wip/reference/install_download_intro.html
2. Installing the probe using the installation package.

The installation package contains the appropriate files for all supported versions of Netcool/OMNIbus. For details about how to install the probe to run with your version of Netcool/OMNIbus, visit the following page on the IBM Tivoli Knowledge Center:

http://www-01.ibm.com/support/knowledgecenter/SSSHTQ/omnibus/probes/all_probes/wip/reference/install_install_intro.html

3. Configuring the probe.

   This guide contains details of the essential configuration required to run this probe. It combines topics that are common to all probes and topics that are peculiar to this probe. For details about additional configuration that is common to all probes, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide*.

Uninstalling Menu Tools

The probe uninstalling process does not remove the menu tools set up in AEL or WebGUI.

To ensure future deployment of the probe is successful, unused tools must be removed manually.

Removing the AEL tools

In the Netool/OMNIbus ObjectServer database, delete the corresponding entry in each table inserted by `UnixHpomTools.sql`.

Removing the WebGUI tools

In **Menu configuration** and **Tools configuration**, delete the item created by `UnixHpomTools.xml`.

Configuration tasks

This section describes the configuration tasks that you need to perform before running the probe.

Configuring ObjectServer acknowledged event triggers

Two triggers are provided in `hpom_automation.sql`, one for the `OPC_MSG_EVENT_ACK` event and one for the `OPC_MSG_EVENT_UNACK` event. If deployed, these triggers update the `Acknowledged` field of all event records with the same `EventId` as these events.

For `OPC_MSG_EVENT_ACK`, the `Acknowledged` field will be set to `1`. For `OPC_MSG_EVENT_UNACK`, the `Acknowledged` field will be set to `0`.

To install the triggers, run the following command:

```
cat $OMNIHOME/probes/<arch>/hpom_automation.sql | $OMNIHOME/bin/nco_sql -server
<server> -user <user> -password <password>
```

To remove the triggers, run the following command:

```
cat $OMNIHOME/probes/<arch>/remove_hpom_automation.sql | $OMNIHOME/bin/nco_sql
-server <server> -user <user> -password <password>
```

Additional probe tools

The probe is supplied with certain additional tools.

Desktop tools

The probe is supplied with the following desktop tools that allow you to display a menu from the desktop to acknowledge or unacknowledge an alert: `UnixHpomTool.sql`, `NHttpHpomProbe.pl`, and `hpom.rules`.

UnixHpomTool.sql

This SQL file creates a UI menu from the desktop that allows you to **Acknowledge** and **Unacknowledge** alerts.

Both menu items call `NHttpHpomProbe.pl` and pass it the following arguments:

- The command to **acknowledge** or **unacknowledge**.
- The `@EventId` from the event.
- The URL to the HTTP command listener of the probe (from the `@URL` field).

For platforms other than linux, the `<arch>` in the Netcool/OMNIbus directories must be updated accordingly.

NHttpHpomProbe.pl

This script is called by the **Acknowledge Alert** and **Unacknowledge Alert** menu items. It sends the command to event factory using the probe's URL.

The command will reach the `hpom.bidir.rules` file to perform an acknowledge or an unacknowledge command accordingly, respectively.

The **Acknowledge Alert** tool sets the event status to `Acknowledged` (1) in the ObjectServer's `alert.status` table.

The **Unacknowledge Alert** tool sets the event status to `Acknowledged` (0) in the ObjectServer's `alert.status` table.

hpom.bidir.rules

This is called in the `hpom.rules` file.

To enable **Acknowledge** and **Unacknowledge** alert operations, uncomment the following line in the `hpom.rules` file:

```
#include "hpom.bidir.rules"
```

Enabling the desktop tools

To enable the desktop tools, using the following steps:

1. For platforms other than Linux, in UnixHpomTool.sql replace linux2x86 with the arch of the platform.
2. Run the following command to create HP Operations Manager menu in Event List:

   ```
   cat $OMNIHOME/probes/<arch>/ UnixHpomTool.sql | $OMNIHOME/bin/nco_sql -
   server <server> -user <user> -password <password>
   ```
3. Uncomment the line `#include "hpom.bidir.rules"` in `hpom.rules`.
4. Run the following command:

   ```
   'chmod +x $OMNIHOME/probes/<arch>/ NHttpHpomProbe.pl'
   ```

Using the desktop tools

The **Acknowledge Alert** and **Unacknowledge Alert** tools issue commands to the event factory through the HTTP port of the probe.

Running the tool has the following results:

- In the ObjectServer, the `Acknowledged` field of the selected event will be set to `1` or `0` for **Acknowledge Alert** and **Unacknowledge Alert**, respectively.
- In the HP Operations Manager system, the corresponding message will be updated to acknowledged or unacknowledged state accordingly.
- The update in HP Operations Manager results in sending an `OPC_MSG_EVENT_ACK` or `OPC_MSG_EVENT_UNACK` event to probe.

WebGUI tools

The probe is supplied with the following WebGUI tools that allow you to create a menu from within the WebGUI to acknowledge or unacknowledge an alert: `UnixHpomTools.xml` and `UnixHpomTools.cgi`.

UnixHpomTools.xml

This file registers the `UnixHpomTools.cgi` script tool in Web GUI. This creates a UI menu called HP Operations Manager that contains two menu items, **Acknowledge Alert** and **Unacknowledge Alert**.

UnixHpomTools.cgi

This script is called by the **Acknowledge Alert** and **Unacknowledge Alert** menu items. It sends the command to the event factory using the probe's URL. The command will reach the `hpom.bidir.rules` file to perform an acknowledge or an unacknowledge command, respectively.

The **Acknowledge Alert** tool sets the event status to `Acknowledged` (1) in the ObjectServer's `alert.status` table.

The **Unacknowledge Alert** tool sets the event status to `Acknowledged` (0) in the ObjectServer's `alert.status` table.

Enabling the WebGUI tools

If the WebGUI is at remote server, copy the tool files from probe package to WebGUI server's `$OMNIHOME/probes/<arch>/` directory.

If HP OM tools already exist, remove the following items from WebGUI:

- **UnixHpomTools** in **CGI Registry**
- **HP Operations Manager** in **Menu Configuration**
- **Acknowlegde_Alert** and **Unacknowledge_Alert** in **Tools Configuration**

To enable the WebGUI tools, use the following steps:

1. Access the command line and run the following command:

   ```
   chmod +x $OMNIHOME/probes/arch/UnixHpomTools.xml
   ```

2. Enter the following command:

   ```
   cp UnixHpomTools.cgi $WEBGUI_HOME/cgi-bin/
   ```

3. Edit `UnixHpomTools.cgi`, to update the following variables: `$OmnihomeDir`, `$Os`, `$OsUser`, `$OsPassword`.
4. Edit `UnixHpomTools.xml`, to update the `fileName` value to the full path of the location from Step 2.
5. Enter the following command:

```
$WEBGUI_HOME/waapi/bin/runwaapi -file $OMNIHOME/probes/arch/
UnixHpomTools.xml -user <user> -password <password>
```

Note : Use a user account that has administration rights.

For more WebGUI configuration information see the following topics:

- Tool Configuration, add user list to **Acknowledge_alert** and **Unknowlegde_alert** tools.
- Menu configuration, add **HP Operations Manager** menu to **Alerts** menu.

Additional manual configuration on Web GUI to enable the HP Operations Manager menu tools

The following steps are a manual alternative to running the **runwaapi** command on `UnixHpomTools.xml`:

1. On the **Tool Configuration** tab, add a group for the **Acknowledge Alert** and **Unacknowledge Alert** tools.

   For **Acknowledge Alert**, make the following settings:

   a. Set the **Type** to `CGI/URL`.
   b. Set the **URL** to `$(SERVER)cgi-bin/UnixHpomTools.cgi?Comand=acknowledge_message`
   c. Set the **Method** to `GET`.
   d. Set **Open In** to **New Window**.
   e. Check both **Execute for each selected row** and **Window for each selected row**.
   f. Under **Access Criteria**, select both the **Netcool_OMNIbus_Admin** and **Netcool_OMNIbus_User** groups.

   For **Unacknowledge Alert**, make the following settings:

   a. Set the **Type** to `CGI/URL`.
   b. Set the **URL** to `$(SERVER)cgi-bin/UnixHpomTools.cgi?Comand=unacknowledge_message`
   c. Set the **Method** to `GET`.
   d. Set **Open In** to **New Window**.
   e. Check both **Execute for each selected row** and **Window for each selected row**.
   f. Under **Access Criteria**, select both the **Netcool_OMNIbus_Admin** and **Netcool_OMNIbus_User** groups.
2. On the **Menu Configuration** tab, add the HP Operations Manager menu to the list of current items.
   a. Highlight **alerts** from the list of **Available menus**.
   b. Click **Modify**.
   c. From the list of **Available items**, highlight the **HP Operations Manager** menu.
   d. Select the **>** button to add **HP Operations Manager** to the list of **Current items**.
   e. Click **Save**.
   f. Verify that the **HP Operations Manager** options now appear when you right-click items on the **alerts** menu.

Troubleshooting WebGUI tools

This troubleshooting topic discusses two commonly encountered problems:

- Tool not appearing in the menu
- CGI script failing to operate correctly

Tool not appearing in the menu

Cause:

- The menu has not been updated with the tools
- The Group setting for the tools (in **Tools Configuration**) and the CGI script (in **CGI Registry**) are not consistent between them, or to the user's group.

CGI script failing to operate correctly

Operational failure may or may not be reported in the WebGUI window, but if the expected result is not observed, you should consider this to be an operational failure, and investigate the following:

- The state of the CGI script in WebGUI environment, for example, file access.
- The state of the message, whether being acknowledged or active in the target system. (Check the HPOM system).
- Whether `nco_http` or `nco_sql` command can be invoked in the CGI script. (Check the access right to the OMNIbus path, the value of OMNIbus directory set in the script, and the environment variables defined for WebGUI runtime, for example, `NCHOME`, `PATH`, and so forth.)
- Whether the `http` command can reach the probe. (Check the probe's log file, and the event record's URL field, try issuing the command manually.)
- Whether the target ObjectServer is reachable from the script command. (Check the user and password, and the object server name in the CGI script.)
- Whether the event field inputs to the script are missing. (Check the tool's configuration, and the related fields in the event list.)

Limitations of the Acknowledge and Unacknowledge alert commands

When probe failover occurs, acknowledging or unacknowledging the alerts originally processed by the counterpart probe will fail. For example, from the slave probe, acknowledging an alert originally processed by the master probe will fail. This is because there will be a mismatched URL value causing the HTTP command to be sent to the inactive destination.

Running the probe

Probes can be run in a variety of ways. The way you chose depends on a number of factors, including your operating system, your environment, and the any high availability considerations that you may have.

For details about how to run the probe, visit the following page on the IBM Tivoli Knowledge Center:

http://www-01.ibm.com/support/knowledgecenter/SSSHTQ/omnibus/probes/all_probes/wip/concept/running_probe.html

Run as `root`

The Probe for HP Operations Manager was developed using the HP API. The API functions must be issued by the user root. OVO allows users with a user ID of zero (`uid 0`), typically `root`, to access the OVO Interface APIs and to define actions for messages that are sent to the management server. Therefore, you can only run the probe as `root`.

If you do not run the probe as root, the probe fails to start and writes the following ProbeWatch message to the log:

```
Must be root to run this probe!
```

Data acquisition

Each probe uses a different method to acquire data. Which method the probe uses depends on the target system from which it receives data.

The Probe for HP Operations Manager acquires event data as it is generated by the HP Operations Manager (HPOM) server. Once processed, the alert is forwarded to the ObjectServer.

The probe makes two connections on separate threads. Both connections are made using the OPC library function `opcif_open()`. The probe connects to two interface instances as follows:

- An instance of the Message Stream Interface (MSI) is connected using the `OPCSVIF_EXTMSGPROC_READ` interface type. This type is used by external message processes.

  Note : The Probe for HP Operations Manager must be installed on the same server as the HPOM server where messages are being configured to be copied to the MSI server.
- An instance of the Message Event Interface (MEI) is connected using the `OPCSVIF_MSG_EVENTS` interface type. This type is used by external message logging and processing tools. The MEI connection receives annotations, acknowledgements done to events in the HPOM database, and notifications of duplicate messages.

Data acquisition is described in the following topics:

- "Configuring HP Operations Manager " on page 8
- "Encrypting the Opcpassword Property" on page 8
- "Processing duplicate events and modify events received from MEI" on page 8
- "Peer-to-peer failover functionality" on page 9

Configuring HP Operations Manager

Before the Probe for HP Operations Manager can successfully acquire event data from the HP Operations Manager server it is necessary to configure HP Operations Manager to send events using the Message Stream Interface (MSI) agent.

Encrypting the Opcpassword Property

The **`Opcuser`** and **`Opcpassword`** properties are specified for MEI only.

The **`Opcpassword`** property must be in an encrypted format. If you have Netcool for Network Node Manager installed, use ovw_crypt. Otherwise, use `nco_aes_crypt`. You can download the latest version of `nco_aes_crypt` from the IBM Software Support web site.

Processing duplicate events and modify events received from MEI

When the Message Event Interface (MEI) receives a duplicate message, the following processing occurs:

1. The probe receives a message `OPC_MSG_EVENT_DUPL_RECEIVED` or `OPC_MSG_EVENT_MODIFY`.
2. The probe sends a query containing the message ID received in the duplicate message to the target system.
3. The target system sends the full message to the probe.
4. The probe sends the message to the ObjectServer for deduplication.

Configuring the Retry and Inactivity properties

The Probe for HP Operations Manager has a timeout facility that allows it to disconnect from the HP Operations Manager server if it does not receive an event within a specified amount of time. To specify how long the probe waits before disconnecting from the target system, set the **`Inactivity`** property.

After the specified length of time, the probe disconnects from the HP Operations Manager server and sends a ProbeWatch message to the ObjectServer. When used with the **`Retry`** property the probe will attempt to reconnect to the target device a number of times before disconnecting. If the **`Retry`** property is set to 0, the probe disconnects after a single period of inactivity.

Peer-to-peer failover functionality

The probe supports failover configurations where two probes run simultaneously. One probe acts as the `master` probe, sending events to the ObjectServer; the other acts as the `slave` probe on standby. If the master probe fails, the slave probe activates.

While the slave probe receives heartbeats from the master probe, it does not forward events to the ObjectServer. If the master probe shuts down, the slave probe stops receiving heartbeats from the master and any events it receives thereafter are forwarded to the ObjectServer on behalf of the master probe. When the master probe is running again, the slave probe continues to receive events, but no longer sends them to the ObjectServer.

Example property file settings for peer-to-peer failover

You set the peer-to-peer failover mode in the properties files of the master and slave probes. The settings differ for a master probe and slave probe.

Note : In the examples, make sure to use the full path for the property value. In other words replace $OMNIHOME with the full path. For example: `/opt/IBM/tivoli/netcool/omnibus`.

The following example shows the peer-to-peer settings from the properties file of a master probe:

```
Server      :    "NCOMS"
RulesFile   :    "master_rules_file"
MessageLog  :    "master_log_file"
PeerHost    :    "slave_hostname"
PeerPort    :    5555 # [communication port between master and slave probe]
Mode        :    "master"
PidFile     : "$OMNIHOME/var/hpom"
```

The following example shows the peer-to-peer settings from the properties file of the corresponding slave probe:

```
Server      :    "NCOMS"
RulesFile   :    "slave_rules_file"
MessageLog  :    "slave_log_file"
PeerHost    :    "master_hostname"
PeerPort    :    5555 # [communication port between master and slave probe]
Mode        :    "slave"
PidFile     : "$OMNIHOME/var/hpom2"
```

Managing the probe over an HTTP/HTTPS connection

IBM Tivoli Netcool/OMNIbus Version 7.4.0 (and later) includes a facility for managing the probe over an HTTP/HTTPS connection. This facility uses the **nco_http** utility supplied with Tivoli Netcool/OMNIbus.

The HTTP/HTTPS command interface replaces the Telnet-based command line interface used in previous version of IBM Tivoli Netcool/OMNIbus.

The following sections show:

- How to configure the command interface.
- The format of the **nco_http** command line.
- The format of the individual probe commands.
- The messages that appear in the log files.
- How to store frequently-used commands in a properties file.

For more information on the HTTP/HTTPS command interface and the utilities it uses, see the chapter on remotely administering probes in the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide*.

Configuring the command interface

To configure the HTTP/HTTPS command interface, set the following properties in the probe's property file:

NHttpd.EnableHTTP: Set this property to True.
NHttpd.ListeningPort: Set this property to the number of the port that the probe uses to listen for HTTP commands.

Optionally, set a value for the following property as required:

NHttpd.ExpireTimeout: Set this property to the maximum elapsed time (in seconds) that and HTTP connection remains idle before it is disconnected.

The *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide* contains a full description of these and all properties for the HTTP/HTTPS command interface.

Messages in the log file

The nco_http utility can make extensive entries in the probe's log file indicating the progress of each operation. These messages can help isolate problems with a request, such as a syntax problem in a command.

To obtain the detailed log information, set the probe's **MessageLevel** property to debug. This enables the logging of the additional information that tracks the progress of a command's execution. For example, the following shows the progress of an **unacknowledge_message** command:

```
Debug: D-NHT-105-002: [HTTP Listener]: Connection from 'hostname'
    (ip-address) on socket 'socket-no'.
Debug: D-NHT-105-008: [HTTP Listener]: Data available from 'hostname'
    (ip-address) on socket 'socket-no'.
Debug: D-NHT-105-009: [WorkPoolThread]: Received HTTP request from 'hostname'
    (ip-address) on socket 'socket-no'.
Information: I-UNK-000-000: Property NHttpd.BasicAuth is empty, allowing
     connection with no authentication
Information: I-UNK-000-000: HTTP: POST: /probe/common request from auth=none
             host=[ip-address]
Debug: D-UNK-000-000: [Bidir: unacknowledge_message] Received $param: [msgId]
```

Storing commands in the nco_http properties file

You can use the **nco_http** utility's properties file ($OMNIHOME/etc/nco_http.props) to hold frequently used command characteristics.

If you have a particular command that you send to the probe regularly, you can store characteristics of that command in the **nco_http** properties file. Once you have done that, the format of the **nco_http** command line is simplified.

You can use the one or more of the following **nco_http** properties to hold default values for the equivalent options on the **nco_http** command line:

Data
DataType
Method
URI

Specify the value of each property in the same way as you would on the command line. Once you have these values in place you do not need to specify the corresponding command line switch unless you want to override the value of the property.

The following is an example of the use of the properties file and the simplification of the **nco_http** command that results. In this example, the **nco_http** properties file contains the following values (note

that line breaks appear for presentational purposes only; when editing the properties use one line for each property value):

```
Data : '{"command":"ackAlarm", "params":[{"alarmId":"alarm1",
"emsId":"EMS1", "managedElementId":"ME1", "username":"root"}]}'
DataType : 'application/JSON'
Method : 'POST'
```

To use this set of values use the following **nco_http** command:

```
$OMNIHOME/bin/nco_http -uri http://test1.example.com:6789
```

Properties and command line options

You use properties to specify how the probe interacts with the device. You can override the default values by using the properties file or the command line options.

The following table describes the properties and command line options specific to this probe. For information about default properties and command line options, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide*.

Table 4. Properties and command line options

Property name	Command line option	Description
FlushBufferInterval *integer*	`-flushbufferinterval` *integer*	Use this property to specify how often (in seconds) that the probe flushes all alerts in the buffer to the ObjectServer. The default is `0` (which instructs the probe to never flush alerts to the ObjectServer).
Inactivity *integer*	`-inactivity` *integer*	Use this property to specify the time (in seconds) the probe waits to receive an event before disconnecting from the target system and sending a ProbeWatch message to the ObjectServer. The default is `0` (The probe does not disconnect .) **Note :** If you do not specify this property the probe will not generate a ProbeWatch message if the target system goes down.

Table 4. Properties and command line options (continued)

Property name	Command line option	Description
MeiInstanceName *string*	`-meiinstancename` *string*	Use this property to specify the instance name when registering with MEI. The instance name will be generated with the "ncoe" prefix followed by the timestamp of the MEI connection. The default is "". **Note :** Defining a fixed instance name allows the probe to reuse an existing instance queue file that was previously registered with HPOM. This applies to peer-to-peer configurations, and will prevent HPOM from sending a disconnect message event.
MsiInstanceName *string*	`-msiinstancename` *string*	Use this property to specify the instance name when registering with MSI. The instance name will be generated with the "nco" prefix followed by the timestamp of the MSI connection. The default is "". **Note :** Defining a fixed instance name will allow the probe to reuse an existing instance queue file that was previously registered with HPOM. This applies for peer-to-peer configurations.
Opcpassword *string*	`-opcpassword` *string*	Use this property to specify the password required with the **Opcuser** property to log in to the system. The default is "".
Opcuser *string*	`-opcuser` *string*	Use this property to specify the username with which the probe logs in to the system. The default is "".

Table 4. Properties and command line options (continued)		
Property name	**Command line option**	**Description**
Retry *string*	-noretry (This is equivalent to **Retry** with a value of false.) -retry (This is equivalent to **Retry** with a value of true.)	Use this property to specify whether the probe attempts to reconnect to the system using a backoff strategy following a reported error. This property takes the following values: false: The probe shuts down and does not attempt to reconnect. true: The probe attempts to reconnect to the system using the backoff strategy, up to a maximum time of 2048 seconds. The default is false.

Elements

The probe breaks event data down into tokens and parses them into elements. Elements are used to assign values to ObjectServer fields; the field values contain the event details in a form that the ObjectServer understands.

This section describes the elements generated by the HP Operations Manager.

- "Annotation Event Alarm Elements" on page 13
- "Message Stream Alarm Elements" on page 14

Annotation Event Alarm Elements

The table describes the elements that the Probe for HP Operations Manager generates for annotation event alarms.

Table 5. Annotation Event Alarm Elements	
Element name	**Element description**
$AnnoId	This element indicates the identifier of the message annotation.
$AnnotationText	This element contains the text of the annotation.
$Author	This element displays the author of the annotation.
$EventFlag	This element displays the event flag name.
$MsgId	This element indicates the unique identifier for the message.
$Text	This element contains the text of the annotation.
$Time	This element indicates the time that the annotation was added.

Message Stream Alarm Elements

The table below describes the elements that the Probe for HP Operations Manager generates for message stream alarms.

Table 6. Message Stream Alarm Elements

Element name	Element description
$Application	This element contains the application name used for the event.
$AutoActionAck	This element contains 0 for the automatic acknowledgment of a successful automatic event; otherwise, it contains 1.
$AutoActionCall	This element indicates the command to issue as an automatic response to the event.
$AutoActionNode	This element indicates the system on which the automatic action should run.
$AutoActionAnnotate	This element contains 1 if HPOM creates start and end annotations for the automatic action event; otherwise, it contains 0.
$CreationTime	This element indicates the time the event was generated.
$DupMsgCounter	This element contains the count of duplicate messages associated with the message stream alarm.
$Group	This element indicates the message group.
$InstructionParameter	This element contains the parameters used to call the external instruction text interface.
$InstructionTag	This element displays the name of the external instruction text interface.
$InstructionType	This element defines whether the internal instruction text or an external interface is used to display instructions for the event.
$MsgId	This element contains a unique identifier for the event.
$MsgKey	This element contains additional message attributes for customized message handling.
$MsgLogOnly	This element contains 1 if the event is automatically inserted into the history event table when it is received by HPOM;otherwise, it contains 0.
$MsgSrc	This element displays the source of the event.
$MsgText	This element displays the event text.

Table 6. Message Stream Alarm Elements (continued)

Element name	Element description
$MsgType	This element displays the type of message contained within the event.
$Node	This element indicates the node name of the server that generated the event.
$Notification	This element contains 1 if HPOM forwards the event to the notification interface; otherwise, it contains 0.
$Object	This element displays the object name used for the event.
$OpActionAck	This element contains 1 if HPOM automatically acknowledges the operator-initiated event after it has finished successfully; otherwise, it contains 0.
$OpActionCall	This element indicates the command to issue as an operator-initiated response to the event.
$OpActionNode	This element indicates the node on which the operator-initiated action should run.
$OpActionAnnotate	This element indicates whether HPOM creates start and end annotations for the operator-initiated action. This element contains 0 if HPOM is not creating start and end annotations; otherwise, it contains 0.
$OrigMsgText	This element displays the original message text.
$ReceiveTime	This element indicates the time the event was received.
$ServiceName	This element displays the name of the service that originated the alarm.
$Severity	This element indicates the severity of the event based on the following event levels: • OPC_SEV_NORMAL= "0" • OPC_SEV_UNKNOWN = "1" • OPC_SEV_WARNING = "2" • OPC_SEV_MINOR= "3" • OPC_SEV_MAJOR = "4" • OPC_SEV_CRITICAL= "5"
$TroubleTicket	This element contains 1 if HPOM forwards the event to the HPOM trouble ticket interface; otherwise, it contains 0.

Table 6. Message Stream Alarm Elements (continued)	
Element name	**Element description**
$TroubleTicketAck	This element contains 1 if HPOM automatically acknowledges the event when it is sent to the trouble ticket interface successfully; otherwise, it contains 0.
$Unmatched	This element contains 0 if the event was forwarded to HPOM because it matched a condition; otherwise, it contains 1.

Custom Message Attribute Elements

The Probe for HP Operations Manager supports the retrieval of custom message attributes which are populated as string elements in addition to the message stream alarm elements.

Each message received has a list of custom attribute names that can be used to retrieve the values for the message. For details about HP Operations Manger custom message attributes see the Administrator Guide supplied with HP Operations Manager.

You will need to customize the Probe for HP Operations Manager rules file to map the custom attributes retrieved to their respective fields in the ObjectServer. If you know what data tokens will be sent in a message, it is best practise to set up corresponding columns for them in the ObjectServer database. with use explicit rules for the column assignments. For details about customizing the rules file, see the *Rules file functions and operators* topic in the *Netcool/OMNibus Probe and Gateway Guide* : http://www-01.ibm.com/support/knowledgecenter/?lang=en#!/SSSHTQ_8.1.0/com.ibm.netcool_OMNIbus.doc_8.1.0/omnibus/wip/probegtwy/reference/omn_prb_rulesfilefncs.html.

To prevent data loss, you can write any unknown tokens to the ExtendedAttr column. For details of how to do this, see the description of the **nvp_add sring** function in the *String functions* topic in the *Netcool/OMNibus Probe and Gateway Guide*: http://www-01.ibm.com/support/knowledgecenter/SSSHTQ_8.1.0/com.ibm.netcool_OMNIbus.doc_8.1.0/omnibus/wip/probegtwy/reference/omn_prb_stringfncs.html?lang=en.

Error messages

Error messages provide information about problems that occur while running the probe. You can use the information that they contain to resolve such problems.

The following table describes the error messages specific to this probe. For information about generic error messages, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide*.

Table 7. Error messages		
Error	**Description**	**Action**
connection to HPOM failed: %s failed to Initialise context : %s\n	The probe cannot connect to HP Operations Manager.	Check that HP Operations Manager is available.
Failed to process event	The probe was unable to process the alert against the rules file.	Contact IBM Software Support.

Table 7. Error messages (continued)

Error	Description	Action

`Failed to determine type of event`
`ERROR failed to process Arguments: %s\n`
`Unexpected type of event: %d`
	The probe found that the event was not in the correct format.	Check that the sender complies with the HP Operations Manager message format.

`Failed to create internal queue`
`Failed to create event: %d`
`failed to set message id: %s", error_msg`
`failed to create message description: %s`
`Failed to set event times : %d"`
`unable to add event to internal event queue`
	There is a problem with the message threading system. This could be due to insufficient memory.	Check that HP Operations Manager is running correctly.
		Make more memory available.

`failed to subscribe for events: %s` `Failed to get message from queue`	The probe cannot get messages from the target system.	Check that HP Operations Manager is running correctly and is available.

Table 7. Error messages (continued)

Error	Description	Action

`failed to create iterator: %s`
`failed to retrieve message ID`
`failed to retreive annotation`
`failed to retrieve annotation time`
`failed to create annotaion container: %s`
`failed to fetch list of annotations: %s`
`failed to create message container: %s`
`failed to create message event container: %s`
`failed to create registration conditions: %s`
`no annotations found for event %s`
`failed to set registration condition description: %s`
`failed to send debug message: %s\n`
`failed to free iterator`
`unable to free registration conditions`
`unable to free message: %s`
	There is a problem with HP Operations Manager communication.	Check that HP Operations Manager is available, and check the MSI configuration.

`error cancelling MEI thread: %d`
`error joinning MEI thread: %d`
`failed to close MEI stream: %s`
`failed to start MEI thread: %s`
`opening of MEI stream failed: %s`
	Failed to interact with the MEI interface for the reason given.	Use the information given in the message to resolve the problem.

Table 7. Error messages (continued)

Error	Description	Action

`error joinning MSI thread: %d`
`error cancelling MSI thread: %d`
`Error: [Message Stream Thread] opening of MSI stream failed: Server is not running`
`failed to start MSI thread: %s`
`failed to close MSI stream: %s`
	Failed to interact with the MSI interface for the reason given.	Use the information given in the message to resolve the problem.

`Failed to get hpom props` `failed to generate props: %s\n`	The probe failed to find the `hpom.props` file.	Check the `hpom.props` file is in the default location. Check the connection to HP Operations Manager.
`Failed to install signal handlers`	The probe was unable to install a signal handler for either a QUIT, INT or TERM. The probe tries to continue; however, any further signal handling results in an error on exit.	Contact IBM Software Support.

ProbeWatch messages

During normal operations, the probe generates ProbeWatch messages and sends them to the ObjectServer. These messages tell the ObjectServer how the probe is running.

The following table describes the raw ProbeWatch error messages that the probe generates. For information about generic ProbeWatch messages, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide*.

Table 8. ProbeWatch messages

ProbeWatch message	Description	Triggers/causes
`failed to Send probewatch %s\n`	The probe failed to send a ProbeWatch message for the reason given.	Check that HP Operations Manager is running correctly and is available.
`Failed to connect - aborting` `Retried up to maximum delay, probe will shutdown`	The probe failed to connect to HP Operations Manager after retrying to connect using the backoff strategy.	Check that HP Operations Manager is running correctly and is available. Check the connection details specified in the `hpom.props` file.

Table 8. ProbeWatch messages (continued)

ProbeWatch message	Description	Triggers/causes
`Must be root to run this probe!!!`	The user running the probe does not have root permissions.	Run the probe as root. For details of how to do this, see the following page in the CORE Netcool/OMNIbus Knowledge Center: https://www.ibm.com/support/knowledgecenter/SSSHTQ_8.1.0/com.ibm.netcool_OMNIbus.doc_8.1.0/omnibus/wip/probegtwy/task/omn_prb_running_probes_suid.html
`not connecting to HPOM as no username and password provided`	The probe failed to connect to HP Operations Manager.	Check the username and password specified.
`Sending probewatch Unable to get events`	The probe cannot get messages from the target system.	Check that HP Operations Manager is running correctly and is available.
`Failure in acknowledging HPOM alert.` `Failure in unacknowledging HPOM alert. Check probe log for details.`	When an `Acknowlegde Alert` or `Unacknowlegde Alert` operation hits exceptions, the probe sends a ProbeWatch message to report the failure. **Note :** The failure must be an outcome from the command executed in probe rules. Other failures due to a mismatched URL or an inactive probe cannot generate this ProbeWatch; the probe is simply unaware of the action.	Check probe log for details.

Appendix A. Notices and Trademarks

This appendix contains the following sections:

- Notices
- Trademarks

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing 2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who want to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department 49XA

© Copyright IBM Corp. 2012, 2018

3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, ibm.com, AIX, Tivoli, zSeries, and Netcool are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Adobe, Acrobat, Portable Document Format (PDF), PostScript, and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.



SC22-5497-05

