IBM® Tivoli® Netcool/OMNIbus Probe for IBM Tivoli NetView
7.0

Reference Guide
March 31, 2011

SC23-9570-02

Note

Before using this information and the product it supports, read the information in Appendix A, “Notices and Trademarks,” on page 11.

Edition notice

This edition applies to version 7.0 of IBM Tivoli Netcool/OMNIbus Probe for IBM Tivoli NetView (SC23-9570-02) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC23-9570-01.

Contents

Document control page

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

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

http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp?topic=/com.ibm.tivoli.nam.doc/welcome_ptsm.htm

Table 1. Document modification history

Document version	Publication date	Comments
SC23-6010-00	October 3, 2006	First IBM publication.
SC23-9570-00	July 25, 2008	Summary section updated.
SC23-9570-01	December 31, 2008	Summary table updated. IPv6 support information added. FIPS information added. Installation section added.
SC23-9570-02	March 31, 2011	Installation section replaced by “Installing probes” on page 2.

Chapter 1. Probe for IBM Tivoli NetView

The IBM Tivoli NetView system is an element manager system which monitors and polls IP devices in a given network. It controls hosts, switches, routers, hubs and all other fundamental network equipment. The probe supports version 7.x of IBM Tivoli NetView.

This guide contains the following sections:

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.

Table 2. Summary	
Probe target	IBM Tivoli NetView version 7.x
Probe executable name	`nco_p_nv7` (UNIX) `nco_p_nv7.exe` (Windows)
Package version	8.0
Probe supported on	For details of supported operating systems, see the following Release Notice on the IBM Software Support Website: https://www-304.ibm.com/support/docview.wss?uid=swg21413178
Properties file	`$OMNIHOME/probes/`*`arch`*`/nv7.props` (UNIX) `%OMNIHOME%\probes\`*`arch`*`\nv7.props` (Windows)
Rules file	`$OMNIHOME/probes/`*`arch`*`/nv7.rules` (UNIX) `%OMNIHOME%\probes\`*`arch`*`\nv7.rules` (Windows)
Requirements	A currently supported version of IBM Tivoli Netcool/OMNIbus. `probe-nco-p-nonnative-base-2_1`
Connection method	API

Table 2. Summary (continued)	
Remote connectivity	No
Multicultural support	Available
Peer-to-peer failover functionality	Available
IP environment	IPv4 and IPv6 **Note :** The probe is supported on IPv6 when running on IBM Tivoli Netcool/OMNIbus 7.4.0.
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*.

Environment variables

Environment variables are specific preset values that establish the probe's working environment with the Tivoli NetView system. From the environment variable specified the probe receives path information of the directory where the library files of the Tivoli NetView system are present.

Before using the probe, you must include the following path in the `$LD_LIBRARY_PATH` environmental variable:

`/opt/OV/lib`

where `/opt/OV` is the location of your Tivoli NetView installation.

Data acquisition

Each probe uses a different method to acquire data. The method that a probe uses is based on the target system from which the probe receives data.

The Probe for IBM Tivoli NetView acquires event data by connecting to a running IBM Tivoli NetView system and capturing traps to form alerts. The probe connects to the trapd process which IBM Tivoli NetView uses. It acquires its events directly from that process.

Note : The probe must be running on the same machine as the IBM Tivoli NetView system. The trapd process must be running before you start the probe.

Internationalization support

The probe supports multibyte character sets (for example, Japanese) and character sets that contain individual multibyte characters (for example German, French, and Spanish). To view the character sets correctly, you must configure the locale settings on the host machine correctly.

If you are using a language that contains multibyte characters, you must set the LANG environment variables to the name of your character set, and export the LC_ALL environment variable. For example, if you are using Japanese, set these environment variables to ja_JP.UTF-8; if you are using German, set these environment variables to de_DE.UTF-8. This will enable the probe to recognise the multibyte characters used by your character set when they occur in any network events.

The probe supports the following language locales:

Table 3. Supported language locales

Languages	**AIX**	**HP-UX**	**Solaris**	**Linux**
English (US)	en_US	en_US	en_US	en_US
Simplified Chinese	zh_CN	zh_CN	zh_CN	zh_CN
Traditional Chinese	zh_TW	zh_TW.eucTW	Zh_TW.big5	zh_TW.big5
Czech	cs_CZ	cs_CZ	cs	cs_CZ
French (standard)	fr_FR	fr_FR	fr	fr_FR
German (standard)	de_DE	de_DE	de	de_DE
Hungarian	hu_HU	hu_HU	hu	hu_HU
Italian (standard)	it_IT	it_IT	it	it_IT
Japanese	ja_JP	ja_JP	ja	ja_JP
Korean	ko_KR	ko_KR	ko	ko_KR
Polish	pl_PL	pl_PL	pl	pl_PL
Portuguese (Brazilian)	pt_BR	pt_BR	pt	pt_BR

Table 3. Supported language locales (continued)				
Languages	**AIX**	**HP-UX**	**Solaris**	**Linux**
Russian	ru_RU	ru_RU	ru	ru_RU
Spanish	es_ES	es_ES	es	es_ES

Example multi-byte character set on Solaris

The following steps describe how to configure Solaris to use the Japanese character set:

1. Install the necessary components for Japanese on to the host machine using the Solaris CD.
2. Set the LANG and LC_ALL environment variables to ja_JP PCK. This uses SJIS encoding.

   Note : You may have to set the LANG in the host machine's default settings file and reboot it to make the changes take effect.
3. Make sure that the file $OMNIHOME/platform/arch/locales/locales.dat has the following entry:

   ```
   locale = ja_JP PCK, japanese, sjis
   ```

   Where ja_JP PCK is the vendor locale, japanese is the Sybase language, and sjis is the Sybase character set.

Example multi-byte configuration on Windows

The following steps describe how to configure Windows to use the Japanese character set:

1. Install the necessary language pack using the **Control Panel**.

   Note : You must reboot the machine to make the character set available.
2. Make sure the file %OMNIHOME%\locales\locales.dat, has the following element:

   ```
   locale = jpn, japanese, sjis
   ```

   Where jpn is the vendor locale, japanese is the Sybase language, and sjis is the Sybase character set

Note : You must reboot the machine to be able to use the probe as a service in the required locale.

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.

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    :    6789 # [communication port between master and slave probe]
Mode        :    "master"
PidFile     : "master_pid_file"
```

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    :    6789 # [communication port between master and slave probe]
Mode        :    "slave"
PidFile     : "slave_pid_file"
```

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 more information about generic Netcool/OMNIbus 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
GetIpStatus *integer*	-getipstatus *integer*	Use this property to specify whether the probe generates the ipStatus element. This element contains the status of the device that generated the trap: 0: The probe does not generate ipStatus elements. 1: The probe generates ipStatus elements. The default is 0. **Note:** For this functionality to work, the probe must be installed on the same host as the IBM Tivoli NetView system.
Heartbeat *integer*	-heartbeat *integer*	Use this property to specify the interval (in seconds) at which the probe sends a ProbeWatch message to the ObjectServer. The default is 60. **Note:** Setting this property to 0 disables the heartbeat functionality.

Table 4. Properties and command line options (continued)

Property name	Command line option	Description
Japanese *integer*	`-japanese` *integer*	Use this property to specify whether the probe displays Kanji characters (Chinese characters used in Japanese writing) in raw token format: `0`: The probe does not display Kanji characters in raw token format. `1`: The probe displays Kanji characters in raw token format. The default is `0`. **Note:** If Kanji characters are expected in the data stream, you must set this property to `1`.
NoName Resolution *integer*	`-nonameresolution` (equivalent to **NoNameResolution** with a value of `1`)	Use this property to specify whether the probe performs name resolution on IP addresses: `0`: The probe performs name resolution. `1`: The probe does not perform name resolution. The default is `0`.
SeqNumFile *string*	`-seqnumfile` *string*	Use this property to specify the path to the log file that the probe uses to record the number of times plus one that it has been started. The default is `$OMNIHOME/var/num_run_seq`.
SNMPMode *integer*	`-V1`(equivalent to **SNMPMode** with a value of `0`) `-V2` (equivalent to **SNMPMode** with a value of `1`)	Use this property to specify whether the probe receives traps from the network node manage in V1 or V2c format: `0`: The probe receives traps in V1 format. `1`: The probe receives traps in V2c format. The default is `0`.
OVAPIDebug *integer*	`-ovapidebug` *integer*	Use this property to specify the level to which the probe sets the NetView API debug. The default is `0`. **Note:** In most cases you should not need to change the value of this property.

Static 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.

The following table describes the static elements that the probe generates:

Table 5. Static elements

Element name	Element description
$community	This element contains the SNMP community string.
$enterprise	This element contains the SNMP enterprise string.
$Eventcount	This element shows the number of event.
$generic-trap	This element contains the SNMP generic trap integer value. Applies only to traps in V1 format.
$IPAddress	This element contains the IP address of the device.
$ipStatus	This element indicates the management status of the device that originated the trap. **Note:** The probe only generates the **ipStatus** element if the **GetIpStatus** property is set to 1.
$LastToken	This element corresponds to the last varbind in the trap.
$Node	This element indicates the node from which the trap originated. **Note:** If the **NoNameResolution** property is set to 0, the probe resolves the host name and sets the $Node element to this value. If the host name is not resolvable by the Domain Name Server, the probe sets the $Node element to the IP address.
$notify	This element contains the SNMP notify value. Applies only to traps in V2 format.
$Sequence	This element indicates the number of times the probe has been run.
$SNMP_Version	This element indicates which version of the SNMP software is monitoring the network.
$Source_Addr	This element shows the IP address of the source host.
$specific-trap	This element contains the SNMP specific trap integer value. Applies only to traps in V1 format.
$uptime	This element shows the amount of time (in 100ths of a second) that the agent sending the trap has been running.

Table 5. Static elements (continued)	
Element name	**Element description**
$UUID	This element contains the unique identifier for an ECS event.

Dynamic elements

The dynamic elements that the probe generates are entirely dependent on the network devices. Each varbind is mapped to elements called $1,$2,$3 and so forth. For each varbind, the object ID is placed in a corresponding element called $OID1, $OID2, $OID3, ... $OIDn where n is the number of varbind elements present.

The probe also generates elements for each varbind element named $raw(1), $raw(2), $raw(3) and so forth; these elements are raw string representations (prior to parsing) of the elements $1,$2,$3 and so forth.

Generic trap handling

Certain devices generate traps of various generic types. How the probe handles each trap depends on its type.

The following table describes the handling of each generic trap type.

Table 6. Generic trap handling	
Generic trap	**Handling**
Generic trap-type 0- Cold Start	Summary field set to Cold Start AlertGroup field set to Generic Severity field set to 4
Generic trap-type 1- Warm Start	Summary field set to Warm Start AlertGroup field set to Generic Severity field set to 4
Generic trap-type 2 - Link Down	Summary field set to Link Down Alert Key set to the $1 varbind (ifIndex) AlertGroup field set to Generic Severity field set to 5 Identifier field set to Node name plus Agent plus generic trap plus specific trap plus ifIndex.
Generic trap-type 3 - Link Up	Summary field set to Link Up Alert Key set to the $1 varbind (ifIndex) AlertGroup field set to Generic Severity field set to 2 Identifier field set to Node name plus Agent plus generic trap plus specific trap plus ifIndex.

Table 6. Generic trap handling (continued)	
Generic trap	**Handling**
Generic trap-type 4 - Authentication	By default, Authentication traps are not discarded.

Generic trap-type 5 - EGP Neighbor Loss	`Summary` field set to `EGP Neighbor Loss`
	`AlertGroup` field set to `Generic`
	`Severity` field set to 3.

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 Netcool/OMNIbus error messages, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide*.

Table 7. Error messages		
Error	**Description**	**Action**
`Could not get the sequence number: Err code` *`number`*	The probe could not retrieve the sequence number and so sets it to `0`.	Check that you have specified a log file using the **`SeqNumFile`** property and that the permissions on the file are set correctly.
`OVerror - Connection to trapd died`	The connection to IBM Tivoli NetView has ended.	Check the status of the trapd process; restart the process if necessary.
`OVerror - No response - request timed out`	A timeout occurred without receiving a response.	Check that the trapd process is running correctly.
`OVerror - Unable to initialize SNMP ov_session - trapd probably died`	The connection to IBM Tivoli NetView could not be initiated.	Check the status of the trapd process; restart the process if necessary.
`Warning: NonnativeProtocol Capabilities: Expected 2, got 0` `Warning: Upgrade nco_p_nonnative: Continuing anyway`	The message refers to functionality not required or used by this probe.	No action is required.

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 ProbeWatch messages that the probe generates. For information about generic Netcool/OMNIbus ProbeWatch messages, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide*.

Table 8. ProbeWatch messages

ProbeWatch message	Description	Triggers or causes
`Heartbeat Message`	This is the regular heartbeat message that the probe sends to the ObjectServer.	The probe sends this message to the ObjectServer with a frequency specified by the **`Heartbeat`** property. This message allows the ObjectServer to confirm that the IBM Tivoli NetView host is still running.

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 wish 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

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.

Other company, product, or service names may be trademarks or service marks of others.

SC23-9570-02