IBM® Tivoli® Netcool/OMNIbus Probe for Nortel Magellan NMS 3.0

Reference Guide
March 31, 2011

SC23-7900-02

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 3.0 of IBM Tivoli Netcool/OMNIbus Probe for Nortel Magellan NMS (SC23-7900-02) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC23-7900-01.

Contents

Document control page

The IBM Tivoli Netcool/OMNIbus Probe for Nortel Magellan NMS 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
01	December 14, 2007	First IBM publication.
02	March 31, 2011	Installation section replaced by "Installing probes" on page 2.

Chapter 1. Probe for Nortel Magellan NMS

The Nortel Magellan NMS is a network management system designed for service providers that provides carrier-grade management solutions. It can be used to manage Nortel Passport and Nortel DPN devices and the networks that those devices support. It also has application interfaces that allow you to integrate it with other systems.

The Probe for Nortel Magellan NMS is a single headed probe that connect to the Nortel Magellan NMS device through a TCP or telnet connection.

The following topics describe the probe and how it works:

- "Summary" on page 1
- "Installing probes" on page 2
- "Data acquisition" on page 2
- "Properties and command line options" on page 5
- "Elements" on page 8
- "Error messages" on page 9
- "ProbeWatch messages" on page 10

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 summarizes the probe.

Table 2. Summary

Probe target	Nortel Magellan NMS
Probe executable file name	`nco_p_nortel_magellan_nms`
Package version	4.0
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=swg21412113
Properties file	`$OMNIHOME/probes/`*`arch`*`/nco_p_nortel_magellan_nms.props`
Rules file	`$OMNIHOME/probes/`*`arch`*`/nco_p_nortel_magellan_nms.rules`
Requirements	A currently supported version of Netcool/OMNIbus
Connection method	TCP/IP
Remote connectivity	The Probe for Nortel Magellan NMS can connect to a remote device. Details of the remote device are specified using the **Host** and **Port** properties in the properties file.

Table 2. Summary (continued)	
Multicultural support	Not Available
Peer-to-peer failover functionality	Available
IP environment	IPv4

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*.

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.

Probe for Nortel Magellan NMS acquires data using the Alarm and Status API. This API is an ASCII interface that provides access to:

- Recent Nortel Magellan NMS alarms (the alarm information is presented in a format that is common to both DPN and Passport devices)
- Current raw state of the DPN and Passport components
- Notification of alarm, status, and rawStateChange events

Data acquisition is described in the following topics:

- "Chat in and chat out scripts" on page 3
- "Data stream capture" on page 3
- "Invalid alarm logging" on page 3
- "Peer-to-peer failover functionality" on page 4

Chat in and chat out scripts

Chat in and chat out scripts control probe login and logout. These scripts are on a single line in the expect-send format (for chat in scripts) or send-expect format (for chat out scripts). You can specify chat in and chat out strings using the **ChatinString** and **ChatoutString** properties or the `-chatinstring` and `-chatoutstring` command line options.

The format is:

```
ChatinString  : expect send expect send....
```

```
ChatoutString : send expect send expect....
```

Note : Each element in the chat strings is separated by white space. In order to send or expect a sequence that includes white space, surround the sequence with single quotes.

A typical chat in script might be:

```
.*login.*:.* anu\r\n .*assword.*:.* anu\r\n
```

The expect text can use any regular expression, while the send text can send any characters, including control characters using the standard UNIX/C escape sequences described in "Escape codes" on page 7.

Raw stream capture

The probe can capture all the raw data stream sent from a device. This raw data is stored in a log file and can be used for debugging, to develop new features for the probe, or to pass to other management systems that require the same data. To enable stream capture, set the **StreamCaptureFile** property to 1 and specify a log file to store the data using the **StreamCaptureFile** property.

Note : The data stream capture function generates a lot of data. When you no longer require data for debugging, set the **StreamCaptureFile** property to 0 to disable the stream capture function.

Invalid alarm logging

You can set the probe to write the details of any received invalid alarms to an invalid log file. To do this, set the **InvalidLogging** property to 1. To specify the name of the file, use the **InvalidAlarmLogFile** property. To specify the maximum size of the log file, use the **MaxInvalidAlarmLog** property.

If the flushtime is activated while the probe is reading an alarm, it discards the buffer. Any incomplete alarms that result following a flushtime activation are logged as invalid. To specify the length of the flushtime period, use the **FlushTime** property.

Data stream capture

The probe can capture the data stream sent from a device in a stream capture file. For each connection, the full data stream is stored in a stream capture file

The data stream is stored using the following naming convention:

streamcapturefile_host_port

where:

- *streamcapturefile* is the value specified by the **StreamCaptureFile** property
- *host* is the name of the host to which the probe is connected
- *port* is the port on which the probe is listening for data

Stream capture data can be used for debugging purposes, to develop new features for the probe, or to pass to other management systems that require the same data.

Note : If you leave the **StreamCaptureFile** property blank, the data stream capture function is disabled.

Rotating stream capture files

The probe can rotate stream capture files; that is, it can write to a stream capture file that is saved and archived periodically when a predefined file size is reached

To use this feature, set the maximum size for the stream capture file using the **MaxStreamCapFileSize** property and set the **DateStreamCapture** property to `1`.

When the **DateStreamCapture** property is set to `1`, the probe creates a stream capture file with the following naming convention:

streamcapturefile_host_port_date_time

By appending the filename with a timestamp, the probe avoids overwriting the old stream capture file.

The maximum file size specified by the **MaxStreamCapFileSize** property acts as an upper limit for the stream capture file. If the probe reads a stream whose size exceeds the remaining allowable space in the current stream capture file, it saves the current file and creates a new one, thus storing the whole stream in a single file.

If you do not specify a maximum size for the stream capture file, it grows indefinitely until the connection is closed. If the **MaxStreamCapFileSize** property is set and the **DateStreamCapture** property is set to `0`, the probe overwrites the stream capture file for that connection each time the maximum file size is reached.

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 information about default properties and command line options, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide,* (SC14-7530).

Table 3. Properties and command line options

Property name	Command line option	Description
`ChatinString` *string*	`-chatinstring` *string*	Use this property to specify the chat in script for connection to the host system. The default is `""`.
`ChatoutString` *string*	`-chatoutstring` *string*	Use this property to specify the chat in script for disconnection from the host system. The default is `""`.
`FlushTime` *integer*	`-flushtime` *integer*	Use this property to specify the time (in seconds) the probe waits before flushing the event to the ObjectServer. The default is 5
`Host` *string*	`-host` *string*	Use this property to specify the name of the host to which the probe connects. The default is `localhost`.
`Inactivity` *integer*	`-inactivity` *integer*	Use this property to specify the time (in seconds) the probe allows a port to be inactive before disconnecting. The default is `0`.

Table 3. Properties and command line options (continued)

Property name	Command line option	Description
InvalidAlarmLogFile *string*	-invalidalarmlogfile *string*	Use this property to specify the path of the file to which the probe logs invalid alarms. The default is $OMNIHOME/var/nortel_ magellan_nms.invalid **Note :** When the probe creates a new invalid alarm log file it appends the date and time to the filename.
InvalidAlarmLogging *integer*	-invalidalarmlogging *integer*	Use this property to specify whether to use invalid alarm logging: 0: The probe does not use invalid alarm logging. 1: The probe uses invalid alarm logging. The default is 0.
MaxInvalidAlarmLogFileSize *integer*	-maxinvalidalarmlog filesize *integer*	Use this property to specify the maximum size (in bytes) for the invalid alarm log file. If the log file exceeds this size, the probe creates a new log file. The default is 1048576.
MaxEvents *integer*	-maxevents *integer*	Use this property to specify the maximum number of events the probe attempts to read before moving on to the next socket. The default is 1.
Port *integer*	-port *integer*	Use this property to specify the port to which the probe connects. The default is 23.
ReadTimeout *integer*	-readtimeout *integer*	Use this property to specify the time (in milliseconds) that the probe waits for a read. The default is 100.

Table 3. Properties and command line options (continued)

Property name	Command line option	Description
ReconnectionAttempts *integer*	-reconnectionattempts *integer*	Use this property to specify the maximum number of times that the probe attempts to reconnect to the socket. The default is 0.
ReconnectionInterval *integer*	-reconnectioninterval *integer*	Use this property to specify the time (in seconds) between successivereconnection attempts. The default is 0.
StreamCaptureFile *string*	-streamcapturefile *string*	Use this property to specify the file the probes uses to store the input stream log. Omitting this property disables the stream capture feature. The default is "".

Escape codes

You can use C-style escape codes in the **ChatinString** and **ChatoutString** properties. This allows you to easily define whether to send escape code sequences after commands.

For example, the following chatin string sends a carriage return character (\r) after the user name and password:

```
ChatinString : ".*: user\r .*: passwd\r .*:"
```

The following table explains the character sequences that are recognized.

Table 4. ChatinString escape codes

Escape code	Character
\b	This escape code specifies the backspace character.
\f	This escape code specifies the form-feed character.
\n	This escape code specifies the new-line character.
\r	This escape code specifies the carriage return character.
\t	This escape code specifies the tab character.
\\	This escape code specifies the backslash character.

Table 4. ChatinString escape codes (continued)

Escape code	Character
`\'`	This escape code specifies the single quote character.
`\"`	This escape code specifies the double quote character.

Note : Due to the way in which the above properties are parsed, the escape sequences for backslash, single quote and double quote must be double-escaped. For example, to send a backslash character (\), use \\\\.

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 elements that the probe generates. Not all the elements described are generated for each event; the elements that the probe generates depend on the event type.

Table 5. Elements

Element name	Element description
`$active`	This element specifies that the event is active.
`$administrativeState`	This element specifies the administrative status of the alarm.
`$alarmRecordId`	This element specifies the identifier of the alarm.
`$alarmType`	This element specifies the type of the alarm.
`$availabilityStatus`	This element specifies the current availability of the equipment.
`$commentData`	This element specifies the proxy alarm generated as a result of SET Alarm.
`$customerId`	This element specifies the identifier of the customer.
`$event`	This element specifies the event.
`$_event_type`	This element specifies type of the event.
`$faultCode`	This element specifies the fault code.
`$fileLineNumber`	This element specifies the file line number of the alarm.
`$fileName`	This element specifies the file name of the alarm.
`$fileVersion`	This element specifies the file version of the alarm.
`$_obj_class`	This element specifies object class.
`$_obj_id`	This element specifies identifier of the object.
`$notificationId`	This element specifies the notification identifier of the alarm.

Table 5. Elements (continued)

Element name	Element description
`$operationalState`	This element specifies the operational status of the element.
`$originatorClass`	This element specifies the originator class of the alarm.
`$probableCause`	This element specifies the probable cause of the alarm.
`$processId`	This element specifies the identifier of the process.
`$_sieve_id`	This element specifies the identifier of the sieve.
`$rawState`	This element specifies the raw state of the alarm.
`$relatedComponents`	This element specifies related components of the alarm.
`$severity`	This element specifies severity of the alarm.
`$standbyStatus`	This element specifies the standby status of the equipment.
`$_time`	This element specifies time of the alarm.

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 6. Error messages

Error message	Description	Action
`CreateAndSet failed for` *Name*	The probe is unable to create an element.	Contact IBM Software Support.

`Event from` *event_source* `did not match the expected format`
`Event from` *Host* `did not match expected format` *Port*
`Failed to find _`*event_type*`: as first 12 characters of buffer`
	The event was not in the correct format.	Check that you are connected to the correct port

`Failed to extract  token name from buffer`	The incoming event did not have the string *_event_type* in it.	Check whether the event stream has got corruputed.
`Failed to process event`	This is an internal error.	Contact IBM Software Support.
`Failed to read from socket - disconnecting`	This is an internal error.	Contact IBM Software Support.
`Invalid Line` *number* `INVALID LINE`	Incoming data may be corrupt.	Check that the device is running properly.

Table 6. Error messages (continued)

Error message	Description	Action
`Read error from socket - disconnecting`	Probe was unable to read a message from the target switch and is disconnecting.	Check that the network management system is working correctly.
`REGEXP: sub` *`number`* `does not exist`	Regular expression has not been specified.	Check the rules file and correct the problem.

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 7. ProbeWatch messages

ProbeWatch message	Description	Triggers or causes
`Event from` *`event_source`* `did not match the expected format`	The event received by the probe does not match any of the supported event formats.	The probe has received an event that does not match any of the supported event formats.

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®, 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.

SC23-7900-02