IBM® Tivoli® Netcool/OMNIbus Exec Probe 4.0

Reference Guide
July 20, 2017

SC27-2302-03

Note

Before using this information and the product it supports, read the information in Appendix A, “Notices and Trademarks,” on page 9.

Edition notice

This edition (SC27-2302-03) applies to version 4.0 of the IBM Tivoli Netcool/OMNIbus Exec Probe and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC27-2302-02.

Contents

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 Exec Probe documentation is provided in softcopy format only. To obtain the most recent version, visit the IBM® Tivoli® Information Center:

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

Table 1. Document modification history

Document version	**Publication date**	**Comments**
SC27-2302-00	August 22, 2008	First IBM publication.
SC27-2302-01	December 31, 2008	Summary table updated. IPv6 support information added. FIPS information added. Installation section added.
SC27-2302-02	November 04, 2011	Information about operating system conventions added in "Conventions used in this guide" on page v. Multicultural support information updated in "Summary" on page 1. Installation section replaced by "Installing probes" on page 2.
SC27-2302-03	July 20, 2017	"Example usage" on page 2 added.

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. Exec Probe

The IBM Tivoli Netcool/OMNIbus Exec Probe is a generic probe that connects to `exec` commands on UNIX and Linux operating systems. It has built in chat in-chat out capabilities and an exposed parser with many definable features, including a termination character, break characters, and message sizing. It supports a wide range of alarm formats.

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.

The following table provides a summary of the Exec Probe.

Table 3. Summary

Probe target	A command or application process.
Probe executable name	`nco_p_exec`
Probe installation package	`omnibus-`*`arch`*`-probe-nco-p-exec-`*`version`*
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=swg21414840
Properties file	`$OMNIHOME/probes/`*`arch`*`/exec.props`
Rules file	`$OMNIHOME/probes/`*`arch`*`/exec.rules`
Requirements	A currently supported version of IBM Tivoli Netcool/OMNIbus.
Connection method	Reads and writes to an internally spawned process using pipes.
Remote connectivity	Not available

Table 3. Summary (continued)

Multicultural support	Available
Peer-to-peer failover functionality	Available
IP environment	IPv4 and IPv6

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*.

Running the probe

The probe is started from the command line.

Before running the probe, use the **ExecCommand** property to specify the command to be spawned.

To start the probe, use the following command:

```
$OMNIHOME/probes/nco_p_exec
```

Example usage

In this example, the Exec Probe is used to connect to a Telnet server to receive an event stream.

Set the following properties in the `exec.props` file:

```
ExecCommand   : 'telnet sol10-build2 8001'
Footer        : 'END JOB'
Header        : 'CNK1'
ParseAsLines  : 1
```

These properties specify that the probe reads events from port 8001 of the Telnet server sol10-build2, treats the text CNK1 as the start of an event and the text END JOB as the end of an event, and treats each line in the event as a separate token.

If the probe receives the following event stream from the Telnet server:

```
CNK1/721TESI001D/CTYCBZ1V1332-P37/013                 99-12-21  18:29:08
7898                                  3073/03999 ARCHIV.CPSPT-10035

CCS7 - BLOCKED TRUNKS

SPC            NETIND  USNAME  FIRST CIC  LAST CIC  NUM OF TRUNKS
--------------+-------+-------+----------+---------+-------------
 4-4- 8-0      NAT1    ISUP      4- 1       4-31    31
END JOB 7898
```

It creates the following tokens:

```
Header:    CNK1/721TESI001D/CTYCBZ1V1332-P37/013            99-12-21  18:29:08
Token0:    7898                                  3073/03999 ARCHIV.CPSPT-10035
Token1:
Token2:    CCS7 - BLOCKED TRUNKS
Token3:
Token4:    SPC            NETIND  USNAME  FIRST CIC  LAST CIC  NUM OF TRUNKS
Token5:    --------------+-------+-------+----------+---------+-------------
Token6:     4-4- 8-0      NAT1    ISUP      4- 1       4-31    31
Footer:    END JOB 7898
```

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 Exec Probe reads events from the internally spawned process specified by the **ExecCommand** property. It connects to the process by opening three pipes: one pipe to the process, one pipe from the process, and one pipe to standard error (stderr). The probe processes events according to the values specified in the properties file.

The **EntrySequence** property specifies the chat in string and the **ExitSequence** property specifies the chat out string.

The probe can be configured to recognize the beginning (header) and the end (footer) of an event. Data can be added to the header and footer of an event using the **Header** and **Footer** properties. The probe can also alter the format of the event and how it is processed using the **TokenSize**, **MessageSize**, and **TerminationCharacter** properties.

Data acquisition is further described in the following topics:

- "Tokenization" on page 3
- "Probe heartbeat" on page 4
- "Peer-to-peer failover functionality" on page 4

Tokenization

You can configure how events are tokenized using the **ParseAsLines** property.

The probe processes the data acquired from a process either one character at a time or one line at a time, depending on the value specified by the **ParseAsLines** property. The probe separates tokens according to the characters specified by the **BreakCharacters** property. Delimiters are specified by the **WhiteSpaces** property.

The following three lines are examples of three events received by the probe from the monitored process:

```
user;host;port; 0;1;2;3;4;this is an example;1
user;host;port; 0;1;2;3;4;this is an example;2
user;host;port; 0;1;2;3;4;this is an example;3
```

The following examples show how the probe parses the events differently depending on the value of the **ParseAsLines** property.

Example 1

If you use the default **ParseAsLines** property value of 0, the probe parses the events one character at a time and separates the tokens as follows:

```
Header = <header>
Token 00 = user;host;port;
Token 01 = 0;1;2;3;4;this
Token 02 = is
Token 03 = an
Token 04 = example;1
Token 05 = user;host;port;
Token 06 = 0;1;2;3;4;this
Token 07 = is
Token 08 = an
Token 09 = example;2
Token 10 = user;host;port;
Token 11 = 0;1;2;3;4;this
Token 12 = is
Token 13 = an
Token 14 = example;3
Footer = <footer>
```

Example 2

If you specify a value of 1 for the **ParseAsLines** property, the probe parses the events one line at a time and separates the tokens as follows:

```
Header = <header>
Token 00 = user;host;port; 0;1;2;3;4;this is an example;1
Token 01 = user;host;port; 0;1;2;3;4;this is an example;2
Token 02 = user;host;port; 0;1;2;3;4;this is an example;3
Footer = <footer>
```

Probe heartbeat

You can specify the interval at which the probe sends a regular heartbeat message to the ObjectServer.

Use the **Heartbeat** property to specify the interval (in seconds) at which the probe sends a heartbeat ProbeWatch message to the ObjectServer. The default is 60 seconds.

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 generic properties and command line options, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide* (SC14-7608).

Table 4. Properties and command line options

Property name	Command line option	Description
BreakCharacters *string*	-break *string*	Use this property to specify the characters that separate non-quoted tokens. The default is ,=\r\n (comma, equals sign, backslash, carriage return, backslash, newline).
EntrySequence *string*	-entrysequence *string*	Use this property to specify the login chat string. The default is "".
ExecCommand *string*	-exec *string*	Use this property to specify the command to be spawned. The default is "".
ExitSequence *string*	-exitsequence *string*	Use this property to specify the logout chat string. The default is "".
Footer *string*	-footer *string*	Use this property to specify the contents of the message footer. The default is "".

Table 4. Properties and command line options (continued)

Property name	Command line option	Description
`Header` *string*	`-header` *string*	Use this property to specify the contents of the message header. The default is `""`.
`Heartbeat` *integer*	`-heartbeat` *integer*	Use this property to specify the interval (in seconds) at which the probe sends a heartbeat ProbeWatch message. The default is `60`.
`MessageSize` *integer*	`-msgbufsize` *integer*	Use this property to specify the maximum number of characters per message output. The default is `2048` characters.
`ParseAsLines` *integer*	`-parseaslines` *integer*	Use this property to specify whether data between the header and footer is passed through a line at a time. The default is `0`.
`QuoteCharacters` *string*	`-quote` *string*	Use this property to specify characters within which event data is processed but not interpreted. The default is `'\'\"` (single quote, backslash, single quote, backslash, double quote).
`TerminationCharacter` *integer*	`-termination` *integer*	Use this property to specify the character that terminates each line of output. The default is `10`.
`TokenSize` *integer*	`-tokensize` *integer*	Use this property to specify the maximum number of characters allowed in any one token. The default is `80`.
`WhiteSpaces` *string*	`-white` *string*	Use this property to specify characters to be ignored, unless they are situated within quotes. The default is ` \t` (space, backslash, tab).

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 Exec Probe generates.

Table 5. Elements

Element name	Element description
`$AlertGroup`	This element indicates the alert status of the probe.
`$Date`	This element contains the date of the event in mm/dd/yy format.
`$Footer`	This element displays the contents of the message footer, if one is specified.
`$Header`	This element displays the contents of the message header, if one is specified.
`$Location`	This element indicates the location of the equipment for which the alarm was raised.
`$Manager`	This element shows the host on which the probe is running.
`$Node`	This element displays the name of the node.
`$Token(01 ... n)`	This element identifies the line that contains the token that was read.
`$Summary`	This element displays the summary of the event.

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* (SC14-7608).

Table 6. Error messages

Error	Description	Action
`Expect call failed!`	The probe cannot connect.	Check the availability of the host.
`Failed to open tty for reading`	The probe was unable to log on.	Check that the properties are set correctly.
`Failed to read ExecCommand Property`	The probe failed to read the **ExecCommand** property.	Check that the **ExecCommand** property is set correctly.

`Footer defined, but no header. Please define or undefine both`
`Header defined, but no footer. Please define or undefine both`
	The probe found that the **Header** and **Footer** properties are undefined.	Check the **Header** and **Footer** properties are set correctly.

Table 6. Error messages (continued)

Error	Description	Action
`Line processing failed`	The probe encountered a tokenization problem and/or parsing error.	Check that the **`TerminationCharacter`**, **`BreakCharacter`**, **`QuoteCharacter`**, and **`WhiteSpaces`** properties have been set correctly.
`LogoutProcedure Failed`	The probe is unable to logout correctly.	Check that the **`ExitSequence`** property is set correctly.
`OplProbeWatch() failed to send XYZ message`	The probe has failed to send mandatory ProbeWatch messages.	Check the connection to the ObjectServer.
`ReadCharacterAtThe Time failed`	The probe failed while reading character.	Contact IBM Support.
`RegComp Failed\n`	The probe received an error while compiling a string (header or footer) into internal exec format and comparing with the set property.	Contact IBM Support.
`select(): reason`	The probe received while setting I/O multiplexing.	Contact IBM Support.

`System Interrupted:`
`System Interruption: reason`
	The probe received a signal while reading character.	Contact IBM Support.

`time: AlertSetFieldValue failed`	The probe was unable to send first and last occurrence fields to the ObjectServer.	Contact IBM 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* (SC14-7608).

Table 7. ProbeWatch messages

ProbeWatch message	Description	Triggers/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.

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.