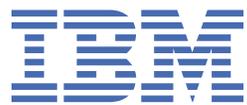IBM® Tivoli® Netcool/OMNIbus Probe for
Samsung Base Station Manager
2.0

Reference Guide
May 25, 2012

IBM

Note

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

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 Probe for Samsung Base Station Manager 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.nam.doc/welcome_ptsm.htm

Table 1. Document modification history		
Document version	**Publication date**	**Comments**
SC23-8845-01	December 31, 2008	Summary table updated. IPv6 support information added. FIPS information added. Installation section added.
SC23-8845-02	March 31, 2011	Installation section replaced by "Installing probes" on page 2.
SC23-8845-03	May 25, 2012	Summary table updated. Information about the event types that the probe processes added to "Data acquisition" on page 2.

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 Samsung BSM

The Probe for Samsung BSM acquires data from the Samsung Radio Access Network (RAN). It is based on the multiheaded socket probe template and connects to the target device through a socket.

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 Probe for Samsung BSM.

Table 3. Summary	
Probe target	Samsung Base Station Manager
Probe executable name	`nco_p_samsung_bsm`
Probe installation package	`omnibus-`*`arch`*`-probe-nco-p-samsung_bsm-`*`version`*
Package version	2.0
Probe supported on	For details of supported operating systems, see the following Release Notice on the IBM Software Support website: http://www.ibm.com/support/docview.wss?uid=swg21591554
Properties file	`$OMNIHOME/probes/`*`arch`*`/samsung_bsm.props`
Rules file	`$OMNIHOME/probes/`*`arch`*`/samsung_bsm.rules`
Requirements	A currently supported version of IBM Tivoli Netcool/OMNIbus
Connection method	Telnet
Remote connectivity	The Probe for Samsung BSM can connect to a device on a remote host. Details of the remote device are specified using the **Host** and **Port** properties in the properties file.
Multicultural support	Available
Peer-to-peer failover functionality	Available

Table 3. Summary (continued)	
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*.

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 Samsung BSM connects to a Samsung Base Station Manager device using the DNS of the device host and the port number through which the host accepts Telnet connections. The DNS and port number values are specified using the **Host** and **Port** properties in the properties file. Once the device is connected to the probe, the alarm events start flowing automatically to the probe and are then parsed to create appropriate tokens. While parsing, the probe reads one event at a time using \030 as the start delimiter, and \031 as the event terminator.

The probe only processes events that are of the following types:

- Normal alarm
- Fault message
- Status message

The probe discards events of all other types.

Data acquisition is described in the following topics:

- "Data stream capture" on page 3
- "Peer-to-peer failover functionality" on page 3

Data stream capture

The probe can capture the data stream sent from a device. This 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 **DateStreamCapture** property to 1. Then specify the name of the stream capture file by setting the **StreamCaptureFile** property or `-streamcapturefile` command line option.

Note : The data stream capture function generates a lot of data. When you no longer require data for debugging, remove the file name specified by the **StreamCaptureFile** property in the properties file.

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).

Property name	Command line option	Description
DateStreamCapture *integer*	`-datestream capture` *integer*	Use this property to specify whether the probe appends a date and time to the stream capture filename. This property takes the following values: 0: Probe does not append the date and time to the stream capture filename 1: Probe appends the date and time to the stream capture filename The default is 0. **Note :** Setting the **DateStreamCapture** property to 1 prevents the stream capture file from being overwritten.
FlushTime *integer*	`-flushtime` *integer*	Use this property to specify the time (in seconds) the probe waits to receive the event terminator before sending the event to the ObjectServer. The default is 5. **Note :** If this property is set to 0, the probe only sends events to the ObjectServer on receipt of an event terminator.
Host *string*	`-host` *string*	Use this property to specify the DNS of the BSM device host. The default is `localhost`. **Note :** You must change the default value to a fixed IP address or host name. For example, a typical IPv6 address would look as follows: `'2001:15f8:106:192:214:5eff:fef5:856'`.

Table 4. Properties and command line options

Table 4. Properties and command line options (continued)

Property name	Command line option	Description
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 (the probe does not disconnect if the port becomes inactive).
InactivityRetry *integer*	`-inactivityretry` *integer*	Use this property to specify the number of consecutive periods of inactivity the probe allows before attempting to reconnect to the interface through the socket. The default is 0 (which disables this feature).
InvalidAlarmLogFile *string*	`-invalidalarm logfile` *string*	Use this property to specify the file to which the probe logs invalid alarms. The current date and time is appended to this filename. The default is `$OMNIHOME/var/ samsung.invalid`.
InvalidAlarmLogging *integer*	`-invalidalarm logging` *integer*	Use this property to specify whether the probe logs invalid alarms. This property takes the following values: 0: Probe does not log invalid alarms 1: Probe logs invalid alarms The default is 0.
MaxInvalidAlarmLogFileSize *integer*	`-maxinvalidalarm logfilesize` *integer*	Use this property to specify the maximum size (in bytes) for the invalid alarm file. When this limit is reached, the probe creates a new file with a new date and time. The default is 1048576.
MaxStreamCapFileSize *integer*	`-maxstreamcap filesize` *integer*	Use this property to specify the maximum size (in bytes) of the stream capture file. When this limit is reached, the probe creates a new file. The default is 0 (infinite). **Note :** If the probe reads a stream whose size exceeds the remaining allowable space in the file, it creates a new one, thus storing the whole stream in a single file.
Port *integer*	`-port` *integer*	Use this property to specify the port to which the probe connects. The default is 23.

Table 4. Properties and command line options (continued)

Property name	Command line option	Description
ReconnectionAttempts _integer_	`-reconnection attempts` _integer_	Use this property to specify the maximum number of times that the probe attempts to reconnect to the socket. The default is 0 (the probe makes unlimited attempts to reconnect to the socket).
ReconnectionInterval _integer_	`-reconnection interval` _integer_	Use this property to specify the time (in seconds) between successive reconnection attempts. The default is 0 (the probe uses standard backoff strategy).
StreamCaptureFile _string_	`-streamcapture file` _string_	Use this property to specify the file that the probe uses to store the input data stream. The default is "".

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 for Samsung BSM generates. Not all the elements described are generated for each event; the elements that the probe generates depends upon the event type.

Table 5. Elements

Element name	Description
`$Clear`	This element specifies the clearance set for the alarm. Note: When the `$Clear` element is set as CLEAR, the Severity of the alarm is automatically set to CLEAR. Else, it is as mentioned in the severity field.
`$Date`	This element specifies the date of the event.
`$FailInfo`	This elements specifies why the alarm failed.
`$Info`	This element specifies information about the alarm.
`$Location`	This element specifies the location of the equipment for which the alarm was raised.
`$MessageId`	This element specifies the tokens that appear in the report header.
`$MessageType`	This element specifies the type of message.
`$Reason`	This element specifies why the alarm occured.
`$Severity`	This element specifies the severity of the alarm.

Table 5. Elements (continued)	
Element name	**Description**
`$Status`	This element specifies the status 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 error messages, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide,* (SC14-7530).

Table 6. Error messages		
Error	**Description**	**Action**
`ConvertEscapes: Error allocating memory for octal code`	The probe was unable to allocate memory for the buffer that contains the event being read. This caused the probe to stop.	Make more memory available.
`CreateAndSet failed for element name`	The probe is unable to create an element.	Contact IBM Software Support.
`Error initializing list Error initializing internal probe list`	This is an internal probe error.	Rerun the probe. If that does not solve the problem, contact IBM Software Support.
`Error reading host file filename`	The probe could not open the hosts file for reading.	Check that the permissions for the hosts file and directories are set correctly.
`Failed to rollover InvalidAlarmLogFile`	The probe was unable to open another alarm log file to write to.	Check that there is sufficient disk space and that the permissions are correct.
`Failed to compile regular expression`	A regular expression is incorrectly formed in the rules file.	Check the rules file for the regular expression and correct the entry.
`Failed to retrieve value for property: property name`	The probe failed to retrieve a value for a required property.	Check that the properties file contains the required property and that it has been specified in the appropriate format.
`Failed to parse text block`	The probe was unable to parse the data specified.	Contact IBM Software Support.
`Failed to allocate memory`	The probe failed to allocate internal storage.	Make more memory available. If this fails, contact IBM Software Support.

Table 6. Error messages (continued)

Error	Description	Action
Have lost all connections - going down	The probe has lost its connection to the socket and is stopping.	Check that the device is running correctly.
Unable to get events	The probe was unable to get any events from the system.	Check that the system is running correctly.
Unable to write to file filename Only managed to write number out of number characters	The probe was unable to write to the file specified.	Check that the permissions of the file are set correctly.

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*, (SC14-7530).

Table 7. ProbeWatch messages

ProbeWatch message	Description	Triggers/causes
Discarding event - Failed to parse	The event is considered invalid as it does not have a line delimiter.	The line delimiter (\n) is missing in the event.
Discarding invalid line between alarms	A line from the alert audit file is not in the correct format.	Check that the Samsung BSM is running correctly.
Discarding on read failure	The probe discarded a partially read event from its buffer following a read failure.	The probe is unable to complete reading an event due to a problem with the connection to the device; for example, the device has dropped the connection to the probe.
Disconnected from system due to shutdown signal	The probe has disconnected from the host.	A shutdown signal was sent to the probe.
Failed to get events	A problem occurred while receiving events.	Either there was a problem initializing the connection due to insufficient memory or (if this message was sent after some events had been parsed) there was a connection failure.
Failed to send Probewatch message ProbeWatch messages	The ProbeWatch messages are not reaching the ObjectServer.	There this a problem either with the probe or the ObjectServer.

Table 7. ProbeWatch messages (continued)		
ProbeWatch message	**Description**	**Triggers/causes**
`Flushtime exceeded while buffering event, flushing event`	There was a problem while buffering and sending the events to the ObjectServer.	The problem occurred because only partial data was received.
`Going Down ...`	The probe is shutting down.	The probe is shutting down after performing the shutdown routine.
`Have lost all connections - going down`	The probe is shutting down because all of its connections have been deactivated.	The number of times that the probe has attempted to reconnect for each of its connections has exceeded that specified by the **ReconnectionAttempts** property.
`Running ...`	The probe is running normally.	The probe has just been started.
`Time (in seconds) without activity before sending an Inactivity ProbeWatch message`	The idle time for which the probe waits for the events when the connectivity is lost.	The time exceeded the idle time for which the probe waits to receive the events.

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

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.

IBM.