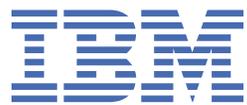IBM® Tivoli® Netcool/OMNIbus Probe for
Kodiak EMS (CORBA)
3.0

Reference Guide
June 12, 2014

IBM

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 (SC23-7665-04) applies to version 3.0 of IBM Tivoli Netcool/OMNIbus Probe for Kodiak EMS (CORBA) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC23-7665-03.

Contents

Document control page

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

The IBM Tivoli Netcool/OMNIbus Probe for Kodiak EMS (CORBA) documentation is provided in softcopy format only. To obtain the most recent version, visit the IBM® Tivoli® Knowledge Center:

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

Table 1. Document modification history		
Document version	**Publication date**	**Comments**
SC23-6091-00	December 4, 2006	First IBM publication.
SC23-7665-00	June 25, 2007	Probe patch and version number updated. **Resynch** property added. Default value for the **OSSId** property changed to `nco_p_kodiak_ems_corba`.
SC23-7665-01	October 31, 2009	Summary table updated. Description for the **ORBLocalHostName** property added to "Properties and command line options" on page 5. Description for **FlushBufferInterval** added.
SC23-7665-02	October 31, 2010	"Summary" on page 1 updated.
SC23-7665-03	March 31, 2011	Installation section replaced by "Installing probes" on page 2. "Firewall considerations" on page 3 added.
SC23-7665-04	June 12, 2014	Summary table updated. Description for the **ORBCharEncoding**, **ORBWCharDefault**, **ProbeMaxEventQueueSize**, and **Resynch** properties added to "Properties and command line options" on page 5. Descriptions for error messages added to "Error messages" on page 8.

Chapter 1. Probe for Kodiak EMS (CORBA)

The Probe for Kodiak EMS (CORBA) manages network entities generated by the Kodiak Real-Time Exchange (RTX) to provide various voice network functions. The Probe for Kodiak EMS (CORBA) acquires data from the Kodiak EMS using a CORBA interface.

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 Kodiak EMS (CORBA).

Table 2. Summary

Probe target	Probe for Kodiak EMS (CORBA) version 5.4
Probe executable name	`nco-p-kodiak-ems-corba`
Package Version	3.0
Probe installation package	`omnibus-`*`arch`*`-probe-nco-p-kodiak-ems-corba-`*`version`*
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=swg21421547
Properties file	`$OMNIHOME/probes/`*`arch`*`/kodiak_ems_corba.props` (for UNIX and Linux operating systems) `%OMNIHOME%\probes\`*`arch`*`\kodiak_ems_corba.props` (for windows operating systems)
Rules file	`$OMNIHOME/probes/`*`arch`*`/kodiak_ems_corba.rules` (for UNIX and Linux operating systems) `%OMNIHOME%\probes\`*`arch`*`/kodiak_ems_corba.rules` (for windows operating systems)

Table 2. Summary (continued)	
Requirements	For details of any additional software that this probe requires, refer to the `description.txt` file that is supplied in its download package.
Connection method	CORBA
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*.

Firewall considerations

When using CORBA probes in conjunction with a firewall, the firewall must be configured so that the probe can connect to the target system.

Most CORBA probes can act as both a server (listening for connections from the target system) and a client (connecting to the port on the target system to which the system writes events). If you are using the probe in conjunction with a firewall, you must add the appropriate firewall rules to enable this dual behavior.

There are three possible firewall protection scenarios, for which you must determine port numbers before adding firewall rules:

1. If the host on which the probe is running is behind a firewall, you must determine what remote host and port number the probe will connect to.
2. If the host on which the target system is running is behind a firewall, you must determine the incoming port on which the probe will listen and to which the target system will connect.
3. If each host is secured with its own firewall, you must determine the following four ports:

   a. The outgoing port (or port range) for the probe.
   b. The hostname and port of the target system.
   c. The outgoing port on which the target system sends events if the probe is running as a client.
   d. The incoming port on which the probe listens for incoming events.

Note : Most, but not all, CORBA probes listen on the port specified by the **ORBLocalPort** property. The default value for this property is 0, which means that an available port is selected at random. If the probe is behind a firewall, the value of the **ORBLocalPort** property must be specified as a fixed port number.

CORBA probes that use EventManager or NotificationManager objects may use different hosts and ports from those that use NamingService and EntryPoint objects. If the probe is configured to get object references from a NamingService or EntryPoint object, you must obtain the host and port information from the system administrator of the target system. When you have this information, you can add the appropriate firewall rules.

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 Kodiak EMS (CORBA) gathers events from the Probe for Kodiak EMS (CORBA) system using a Common Object Request Broker Architecture (CORBA) interface. CORBA is an Object Management Group specification that provides a standard interface definition between objects in a distributed environment; that is, it allows applications to communicate with one another regardless of where they are located or who has designed them.

Data acquisition is described in the following topics:

Connecting to the CORBA interface

The following steps describe how the probe acquires data from the Kodiak EMS:

1. The probe starts up and initializes an ORB.
2. The probe reads the following URL to locate the Kodiak EMS:

```
http://ip_address:9090/ior/ior.txt
```

Where *ip_address* is the IP address of the machine on which the Kodiak EMS is running.

3. The probe uses the IOR to connect to the naming service and connects to the `AlarmManager` object which is bound to the name `KODIAK_EMS_ALARMS_1`.

4. The probe calls the `AlarmManager getAlarms()` method to synchronize with the Kodiak EMS.

5. The probe generates alerts and sends them to the ObjectServer.

6. The probe creates a `NotificationManager` instance and registers it with the `AlarmManager` reference.

7. The probe waits for unsolicited alarms from the simulator and converts them into alerts.

Failover

The Probe for Kodiak EMS (CORBA) can failover to a standby Kodiak EMS server, if the primary Kodiak EMS server becomes inactive. By default, the probe connects to the primary server set in the **TargetEMSIPAddr** property. If this target server becomes inactive then the probe would failover to the standby server set in the **StandbyEMSIPAddr** property.

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
EMSBindName *string*	`-emsbindname` *string*	Use this property to specify the bind name that the Kodiak EMS uses for the `AlarmManager` instance from which the probe acquires alerts. The default is KODIAK_EMS_ALARM_1.
EMSHeartBeat Interval *integer*	`-emsheartbeatinterval` *integer*	Use this property to specify the interval (in seconds) that the probe waits between sending successive heartbeats to the Kodiak EMS. The default is 60.
EMSPortNumber *integer*	`-emsportnumber` *integer*	Use this property to specify the port number within the Kodiak EMS on which the probe listens for alerts. The default is 9090.
FlushBufferInterval *integer*	`-flushbufferinterval` *integer*	Use this property to specify how often (in seconds) the probe flushes all alerts in the buffer to the ObjectServer. The default is 0 (which instructs the probe to never flush the alerts to the ObjectServer).
ORBCharEncoding *string*	`-orbcharencoding` *string*	Use this property to specify the native character encoding set used by the Object Request Broker (ORB) for character data. The default is UTF8.
ORBDebug *integer*	`-orbdebug` *integer*	Use this property to specify the debug level used by the ORB. The default is 0.
ORBLocalHostName *string*	`-orblocalhostname` *string*	Use this property to specify the name of the host machine on which the ORB is running. The probe writes this name to the IOR file that it then sends to the EMS. The default is "".

Table 3. Properties and command line options (continued)

Property name	Command line option	Description
ORBLocalPort *integer*	`-orblocalport` *integer*	Use this property to specify the local port to which the Object Request Broker (ORB) listens. The default is 0 (ORB selects an available port at random).
ORBWCharDefault *string*	`-orbwchardefault` *string*	Use this property to specify what wide character (wchar) set the IBM ORB uses when communicating with other ORBs that do not publish a wchar set. The default is UTF16.
OSSId *string*	`-ossid` *string*	Use this property to specify the unique identifier that the probe instance will be known as within the Kodiak EMS. The default is nco_p_kodiak_ems_corba.
ProbeMaxEventQueueSize *integer*	`-probemaxeventqueue size` *integer*	Use this property to specify the maximum queue size of the probe's internal event queue. The default is 10000
Resynch *string*	`-noresynch` (This is equivalent to **Resynch** with a value of `false`.) `-resynch` (This is equivalent to **Resynch** with a value of true.)	Use this property to specify whether the probe attempts to resynchronize existing alarms before retrieving new alarms. This property takes the following values: `false` : The probe does not attempt to resynchronize the alarms. `true` : The probe attempts to resynchronize the alarms. The default is `false`.
Retry *integer*	`-retry` *integer*	Use this property to specify the number of times the probe retries connecting to the host system before shutting down. The default is 1.
RetryWaitInterval *integer*	`-retrywaitinterval` *integer*	Use this property to specify the time (in seconds) between successive retry attempts. The default is 0.
SleepDuration *integer*	`-sleepduration` *integer*	Use this property to specify the period (in seconds) the probe remains asleep until the arrival of a new alarm. The default is 5.

Table 3. Properties and command line options (continued)		
Property name	**Command line option**	**Description**
StandbyEMSIPAddr *string*	`-standbyemsipaddr` *string*	Use this property to specify the IP address of the standby Kodiak EMS. The default is `127.0.0.1`.
TargetEMSIPAddr *string*	`-targetemsipaddr` *string*	Use this property to specify the IP address of the target Kodiak EMS. The default is `127.0.0.1`.

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 Kodiak EMS (CORBA) generates. Not all the elements described are generated for each event; the elements that the probe generates depends upon the event type.

Table 4. Elements	
Element name	**Element description**
`$AlarmCode`	This element identifies the alarm type.
`$AlarmGroup`	This element identifies the group to which the alarm belongs. Possible values are: • EQUIPMENT • ENVIRONMENTAL
`$AlarmSeverity`	This element identifies the severity of the alarm as reported by the EMS.
`$FirstOccurrence`	This element identifies the first occurrence of an event. It is based on the `SourceTime` element of the alarm.
`$LastOccurrence`	This element identifies the last time the event was modified. It is based on the `UpdateTime` element of the alarm.
`$ProbableCause`	This element displays a textual description of the probable cause.
`$ServiceAffecting`	This element indicates whether the alarm is service affecting.
`$SourceName`	This element identifies the source that sent 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 5. Error messages		
Error	**Description**	**Action**
Event queue full. Fail to add alarm into queue.	The probe failed to add the new event into the event queue because it is full. This may occur when the probe was not able to catch up with the event rate received from the Kodiak EMS or there was an error removing events from the queue. This may cause a potential event loss.	Check the probe log to see whether an error occurred during event processing or while sending the alert to the ObjectServer. Check whether there was an event flood from the Kodiak EMS.
Failed to add the ipAddress to the list	The probe failed to add the IP address to the list of EMS systems to which it should connect.	Check that the IP address is specified correctly in the **TargetEMSIPAddr** property.
No alarm manager interface	The probe could not obtain the AlarmManager interface instance on the EMS card.	Check that the Kodiak EMS is running correctly.
Malformed URL for target for IP address: *ip_address*	The probe generated a malformed name server locator URL.	Check to see if the formats of the **TargetEMSIPAddr** and **StandbyEMSIPAddr** properties are correct.
Input Output Exception for IP address: *ip_address*	An error occurred when attempting to retrieve the naming service IOR.	Check that the ior.txt file is available on the target host. Check that the port number is specified correctly in the **EMSPortNumber** property.
ParseAlarmInformation: Failed to send event : *java_exception* ParseAlarmInformation: Failed to send alarm operation update : *java_exception*	Probe failed to parse the alarm received from the Kodiak EMS system. The data received may be corrupt.	Check that the Kodiak EMS is running correctly. Check that the alarm conforms to the AlarmInformation structure.
CORBA.TRANSIENT exception raised. Lost connection to EMS!	The probe has lost connection to the EMS system. The EMS is not live.	Check the EMS server availability. Check the connection between the probe and the EMS.

Table 5. Error messages (continued)

Error	Description	Action
`Problem getting alarm during resynch.`	The probe failed to call the `getAlarms` function with the OSS ID specified.	Check that the EMS system received the `getAlarms` request during the probe resynchronization phase. Check the value set for the **OSSId** property.
`Could not obtain alarm interface`	The probe failed to narrow the object reference to the interface object.	This is a CORBA related error message. Check that the CORBA installation is correctly configured.
`Probe shutting down since it cannot connect to the EMS server.`	The probe failed to connect to the EMS server during startup.	Check the EMS server availability.

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.

IBM®