

IBM® Tivoli® Netcool/OMNIBus Probe for ZTE
NetNumen N31 (CORBA)
2.0

Reference Guide
March 02, 2012



Notice

Before using this information and the product it supports, read the information in [Appendix A, “Notices and Trademarks,”](#) on page 17.

Edition notice

This edition (SC27-2702-03) applies to version 2.0 of IBM Tivoli Netcool/OMNIbus Probe for ZTE NetNumen N31 (CORBA) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC27-2702-02.

© **Copyright International Business Machines Corporation 2009, 2012.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this guide.....	v
Document Control Page.....	v
Conventions used in this guide.....	v
 Chapter 1. Probe for ZTE NetNumen N31 (CORBA).....	1
Summary.....	1
Installing probes.....	2
Firewall considerations.....	3
Data acquisition.....	3
Connecting to the CORBA interface.....	4
Retrieving alarms and notifications.....	4
Filtering alarms and notifications.....	4
Lookup table.....	5
Timeout.....	6
Backoff strategy.....	6
Command line interface.....	6
Peer-to-peer failover functionality.....	7
Properties and command line options.....	8
Error messages.....	11
ProbeWatch messages.....	14
Running the probe.....	14
Running multiple probes.....	15
 Appendix A. Notices and Trademarks.....	17
Notices.....	17
Trademarks.....	18

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 ZTE NetNumen N31 (CORBA) documentation is provided in softcopy format only. To obtain the most recent version, visit the IBM Documentation site:

https://www.ibm.com/support/knowledgecenter/SSSHTQ_int/omnibus/probes/common/Probes.html

Table 1. Document modification history		
Document version	Publication date	Comments
SC27-2702-00	June 20, 2009	First IBM publication.
SC27-2702-01	October 31, 2010	“Summary” on page 1 updated. Description for FlushBufferInterval added.
SC27-2702-02	March 31, 2011	Installation section replaced by “Installing probes” on page 2. “Firewall considerations” on page 3 added.
SC27-2702-03	March 02, 2012	Information about operating system conventions added in “Conventions used in this guide” on page v. Requirements updated in “Summary” on page 1. Information about connecting to the CORBA interface was updated in “Connecting to the CORBA interface” on page 4. The following properties were added in “Properties and command line options” on page 8: • ORBCharEncoding • ORBLocalHost • ORBWCharDefault

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 ZTE NetNumen N31 (CORBA)

The ZTE NetNumen N31 Fixed Network Management System (NMS) is an integrated network operation and maintenance system. It controls different interconnected networks and performs integrated network fault management, fault location, and performance analysis.

The Probe for ZTE NetNumen N31 (CORBA) reads events from the ZTE N31 NMS using a Common Object Request Broker Architecture (CORBA) interface.

The probe supports the following ZTE network elements:

- N31 (NGN)
- N31 (IP)
- N31 (AN)
- N31 (VAS)

This probe is described in the following sections:

- [“Summary” on page 1](#)
- [“Installing probes” on page 2](#)
- [“Firewall considerations” on page 3](#)
- [“Data acquisition” on page 3](#)
- [“Properties and command line options” on page 8](#)
- [“Error messages” on page 11](#)
- [“ProbeWatch messages” on page 14](#)
- [“Running the probe” on page 14](#)

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 ZTE NetNumen N31 (CORBA).

Table 3. Summary	
Probe target	ZTE N31 NMS version 4.0.1
Probe executable name	nco_p_zten31_fixednms_corba
Probe installation package	omnibus-arch-probe-nco-p-zten31-fixednms-corba-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: https://www-304.ibm.com/support/docview.wss?uid=swg21579034
Properties file	\$OMNIHOME/probes/arch/nco_p_zten31_fixednms_corba.props
Rules file	\$OMNIHOME/probes/arch/nco_p_zten31_fixednms_corba.rules

Table 3. Summary (continued)	
Requirements	One of the following versions of IBM Tivoli Netcool/OMNIBus: • V7.2.0 with Interim Fix 4 (7.2.0.7-IF0004) • V7.2.1 with Interim Fix 3 (7.2.1.5-IF0003) • V7.3.0 • V7.3.1 Nonnative Probe Server: probe-nonnative-base-14 Command Port library: probe-command-port-3 IBM® Object Request Broker (IBM ORB, supplied with Netcool/OMNIBus) Java™ Runtime Environment (JRE) 1.5
Connection method	CORBA
Remote connectivity	The probe can connect to a remote device using the CORBA interface.
Multicultural support	Available
Peer-to-peer failover functionality	Available
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 <i>IBM Tivoli Netcool/OMNIBus Installation and Deployment Guide</i>.

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 IBM Documentation:

https://www.ibm.com/support/knowledgecenter/SSSHTQ_int/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 in IBM Documentation:

https://www.ibm.com/support/knowledgecenter/SSSHTQ_int/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

The probe gathers events from the ZTE N31 NMS using a CORBA interface.

Data acquisition is further described in the following topics:

- [“Connecting to the CORBA interface” on page 4](#)
- [“Retrieving alarms and notifications” on page 4](#)
- [“Filtering alarms and notifications” on page 4](#)
- [“Lookup table” on page 5](#)
- [“Timeout” on page 6](#)
- [“Backoff strategy” on page 6](#)
- [“Command line interface” on page 6](#)
- [“Peer-to-peer failover functionality” on page 7](#)

Connecting to the CORBA interface

The probe acts as an Integration Reference Point (IRP) Manager and connects to the ZTE N31 NMS system using a CORBA interface.

The probe uses the value of the **ServiceObjectStringRef** property to obtain the string reference of the ServiceMng object from the Naming Service. The probe then uses the ServiceMng object to acquire references on the Alarm IRP and Notification IRP objects specified by the **AlarmObjectName** and **NotificationObjectName** properties.

The probe uses the Alarm IRP object for alarm resynchronization and the Notification IRP object to subscribe to realtime events.

Retrieving alarms and notifications

If the **Resynch** property is set to true, the probe initially receives a list of all active alarms from the AlarmIRP object.

The probe then connects to the NotificationIRP object and uses the CORBA notification push model to receive new alarms. If the **Resynch** property is set to false, the probe only receives new alarms.

Filtering alarms and notifications

You can use the **AlarmFilter** and **NotificationFilter** properties to filter the events that the probe gathers from the ZTE N31 NMS.

When you use these properties, you must use the actual token names. For example, the token PERCEIVED_SEVERITY represents the element NV_PERCEIVED_SEVERITY.

To specify that the probe gathers only those notifications with a perceived severity of 3, set the **NotificationFilter** property to `$PERCEIVED_SEVERITY == 3`.

You can specify more complex filters using AND, OR, and NOT statements. For example, to specify that the probe gathers notifications with a perceived severity of 3 or 4, set the **NotificationFilter** property to the following value:

```
($PERCEIVED_SEVERITY == 3) OR ($PERCEIVED_SEVERITY == 4)
```

The following table lists the token mappings available for use with the **AlarmFilter** and **NotificationFilter** properties.

Table 4. Token mappings	
Element	Token
NV_NETYPE	NETYPE
NV_NEIP	NEIP
NV_EVENT_TIME	EVENT_TIME
NV_PROBABLE_CAUSE	PROBABLE_CAUSE
NV_PERCEIVED_SEVERITY	PERCEIVED_SEVERITY
NV_ENTITY_ID	ENTITY_ID
NV_ALARM_ID	ALARM_ID
NV_NOTIFICATION_ID	NOTIFICATION_ID

Table 4. Token mappings (continued)	
Element	Token
NV_CLEAR_USER_ID	CLEAR_USER_ID
NV_CLEAR_SYSTEM_ID	CLEAR_SYSTEM_ID
NV_ACK_USER_ID	ACK_USER_ID
NV_ACK_SYSTEM_ID	ACK_SYSTEM_ID
NV_ACK_STATE	ACK_STATE
NV_COMMENTS	COMMENTS
NV_ALARM_CLEARED_TIME	ALARM_CLEARED_TIME
NV_ALARM_CHANGED_TIME	ALARM_CHANGED_TIME
NV_ACK_TIME	ACK_TIME
NV_EVENT_CODE	EVENT_CODE
NV_EVENT_DESC	EVENT_DESC
NV_PROBABLE_CAUSE_CODE	PROBABLE_CAUSE_CODE
NV_PROBABLE_CAUSE_DESC	PROBABLE_CAUSE_DESC
NV_ADDITIONAL_INFORMATION	ADDITIONAL_INFORMATION
NV_POSITION	POSITION
NV_INFO	INFO
NV_ADDITIONAL_TEXT	ADDITIONAL_TEXT

Lookup table

The probe is supplied with a lookup table that contains details of the various types of alarms that the ZTE system generates.

At installation, the `Corba_zten31_fixednms.lookup` file supplied with the probe installation package is installed to the following location:

`$OMNIHOME/probes/includes/`

This file is referenced by the following line in the rules file:

```
include "../includes/Corba_zten31_fixednms.lookup"
```

Note: The include command assumes that the probe is run from the standard `$OMNIHOME/probes/` directory. If you are running the probe from a different directory, replace `".."` with the absolute directory path to the lookup table. Do not use the `$OMNIHOME` environment variable in this directory path.

Timeout

The probe has a timeout facility that allows it to disconnect from the system if there has been no activity for a predefined amount of time.

You can use the **Timeout** property to specify how long the probe waits before disconnecting. The default is 0, which instructs the probe to maintain the connection indefinitely.

Backoff strategy

If the **Retry** property is set to true, and the probe fails to establish a connection or loses an existing connection to the device, the probe reverts to a backoff strategy.

The probe's backoff strategy is to try to reestablish a connection at successive intervals of one second, two seconds, four seconds, eight seconds, and so on, up to a maximum of 4096 seconds. When the maximum retry interval is reached, the probe stops trying to reconnect and will not try again until it is restarted.

Command line interface

The probe is supplied with a Command Line Interface (CLI). This interface enables you to execute commands to acknowledge alarms or request a full resynchronization.

To use the CLI, you must use the **CommandPort** property to specify a port through which commands will be sent. The default port is 6970. When you want to issue commands, use Telnet to connect through this port. You can use the **CommandPortLimit** property to limit the number of Telnet connections that the probe can make at one time.

Note: If there is a firewall between the probe and the ZTE N31 NMS, keep the following ports open: TCP 21115, TCP 21126, and TCP 21111.

The following table describes the commands that you can use with the CLI.

Table 5. CLI commands	
Command	Description
exit/quit	Use this command to close the connection.
help	Use this command to display online help about the CLI.
resynch_all	Use this command to perform a full resynchronization with the CORBA interface.
resynch_filter <i>filter</i>	Use this command to perform partial resynchronization with the CORBA interface. Note: The <i>filter</i> parameter uses the same filter format as the AlarmFilter property.
acknowledge_alarm <i>alarm_id user_id</i>	Use this command to acknowledge an alarm in the CORBA interface by specifying the NV_ALARM_ID identifier of the alarm and the user identifier of the user acknowledging the alarm. Before you can use this command, you must specify a value for the AckSystemId property.

Table 5. CLI commands (continued)	
Command	Description
unacknowledge_alarm <i>alarm_id user_id</i>	Use this command to unacknowledge an alarm in the CORBA interface by specifying the NV_ALARM_ID identifier of the alarm and the user identifier of the user acknowledging the alarm. Before you can use this command, you must specify a value for the AckSystemId property.
clear_alarm <i>alarm_id user_id</i>	Use this command to clear an alarm by specifying the NV_ALARM_ID identifier of the alarm and the user identifier of the user acknowledging the alarm. Before you can use this command, you must specify a value for the ClearSystemId property.
comment_alarm <i>alarm_id user_id commentText</i>	Use this command to add comments to an alarm by specifying the NV_ALARM_ID identifier of the alarm, the user identifier of the user making the comment, and the text of the comment. Before you can use this command, you must specify a value for the CommentSystemId property.
version	Use this command to display the version of the probe.

CLI scripts

Because the CLI uses Telnet connections, you can connect to the probe from anywhere by creating a desktop tool to open a Telnet connection, send a command, and then close the connection. This means that simple scripts can be set up to allow users to acknowledge selected events from the IBM Tivoli® Netcool/OMNIbus event list.

Peer-to-peer failover functionality

The probe supports failover configurations where two probes run simultaneously. One probe acts as the master probe, sending events to the ObjectServer; the other acts as the slave probe on standby. If the master probe fails, the slave probe activates.

While the slave probe receives heartbeats from the master probe, it does not forward events to the ObjectServer. If the master probe shuts down, the slave probe stops receiving heartbeats from the master and any events it receives thereafter are forwarded to the ObjectServer on behalf of the master probe. When the master probe is running again, the slave probe continues to receive events, but no longer sends them to the ObjectServer.

Example property file settings for peer-to-peer failover

You set the peer-to-peer failover mode in the properties files of the master and slave probes. The settings differ for a master probe and slave probe.

Note: In the examples, make sure to use the full path for the property value. In other words replace \$OMNIHOME with the full path. For example: /opt/IBM/tivoli/netcool.

The following example shows the peer-to-peer settings from the properties file of a master probe:

```
Server      : "NCOMS"
RulesFile   : "master_rules_file"
```

```

MessageLog : "master_log_file"
PeerHost   : "slave_hostname"
PeerPort   : 6789 # [communication port between master and slave probe]
Mode       : "master"
PidFile    : "master_pid_file"

```

The following example shows the peer-to-peer settings from the properties file of the corresponding slave probe:

```

Server      : "NCOMS"
RulesFile   : "slave_rules_file"
MessageLog  : "slave_log_file"
PeerHost    : "master_hostname"
PeerPort    : 6789 # [communication port between master and slave probe]
Mode        : "slave"
PidFile     : "slave_pid_file"

```

Properties and command line options

You use properties to specify how the probe interacts with the device. You can override the default values by using the properties file or the command line options.

The following table describes the properties and command line options specific to this probe. For more information about generic Netcool/OMNIbus properties and command line options, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide*.

Table 6. Properties and command line options		
Property name	Command line option	Description
AckSystemId <i>string</i>	-acksystemid <i>string</i>	Use this property to specify the system identifier to use when acknowledging alarms using the CLI. The default is " ". Note: You must provide the system ID of the processing system instead of the default null value.
Agentheartbeat <i>integer</i>	-agentheartbeat <i>integer</i>	Use this property to specify the frequency (in seconds) with which the probe checks the status of the ZTE system. The default is 300.
AlarmFilter <i>string</i>	-alarmfilter <i>string</i>	Use this property to specify the filter that the Alarm IRP uses to limit the alarms sent to the probe. The default is " ".
AlarmObjectName <i>string</i>	-alarmobjectname <i>string</i>	Use this property to specify the name that the probe uses to resolve the Alarm IRP object in the Naming Service. The default is alarm.

Table 6. Properties and command line options (continued)

Property name	Command line option	Description
ClearSystemId <i>string</i>	-clearsystemid <i>string</i>	Use this property to specify the system identifier to use when clearing alarms using the CLI. The default is " ".
CommandPort <i>integer</i>	-commandport <i>integer</i>	Use this property to specify the port through which you will send commands using the CLI. The default is 6970.
CommandPortLimit <i>integer</i>	-commandportlimit <i>integer</i>	Use this property to specify the maximum number of Telnet connections that the probe can make using the command port. The default is 10.
CommentSystemId <i>string</i>	-commentssystemid <i>string</i>	Use this property to specify the system identifier to use when adding comments to an alarm using the CLI. The default is " ".
FlushBufferInterval <i>integer</i>	-flushbufferinterval <i>integer</i>	Use this property to specify how often (in seconds) the probe flushes all alerts in the buffer to the ObjectServer. The default is 0 (the probe does not flush alerts to the ObjectServer).
NotificationFilter <i>string</i>	-notificationfilter <i>string</i>	Use this property to specify the filter that the Notification IRP uses to limit the notifications sent to the probe. The default is " ".
NotificationObjectName <i>string</i>	-notificationobjectname <i>string</i>	Use this property to specify the name that the probe uses to resolve the Notification IRP object in the Naming Service. The default is NOTIFICATIONIRP.
ORBCharEncoding <i>string</i>	-orbcharencoding <i>string</i>	Use this property to specify the native character encoding set used by the Object Request Broker (ORB) for character data. The default is UTF8.

Table 6. Properties and command line options (continued)

Property name	Command line option	Description
ORBLocalHost <i>string</i>	-orblocalhost <i>string</i>	Use this property to specify the host name or IP address of the local ORB host. The default is "".
ORBLocalPort <i>integer</i>	-orblocalport <i>integer</i>	Use this property to specify the local port to which the ORB listens. The default is 0 (which instructs the ORB to select an available port at random).
ORBWCharDefault <i>string</i>	-orbwchardefault <i>string</i>	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.
Resynch <i>string</i>	-resynch (This is equivalent to Resynch with a value of true.) -noresynch (This is equivalent to Resynch with a value of false.)	Use this property to specify whether the probe requests all active alarms from the ZTE system before connecting to the notification service for new alarms. This property takes the following values: false: The probe does not perform resynchronization. true: The probe performs resynchronization. The default is false.
Retry <i>string</i>	-retry (This is equivalent to Retry with a value of true.) -noretry (This is equivalent to Retry with a value of false.)	Use this property to specify whether the probe attempts to reconnect to the ZTE system following a timeout. This property takes the following values: false: The probe does not attempt to reconnect to the system. true: The probe attempts to reconnect to the system. The default is false.

Table 6. Properties and command line options (continued)

Property name	Command line option	Description
ServiceObjectStringRef <i>string</i>	-serviceobjectstringref <i>string</i>	Use this property to specify the string reference of the ServiceMng object registered with the Naming Service. The default is corbaname:iiop: <i>NamingServiceHost</i> : <i>NamingServicePort</i> / StandardImplName/Naming/ root#service, where <i>NamingServiceHost</i> is the IP address of the Naming Service host and <i>NamingServicePort</i> is the port number on which the Naming Service listens for connections.
Timeout <i>integer</i>	-timeout <i>integer</i>	Use this property to specify the time period (in seconds) that the probe waits for the next alarm before disconnecting. The default is 0 (the session remains open indefinitely).

Error messages

Error messages provide information about problems that occur while running the probe. You can use the information that they contain to resolve such problems.

The following table describes the error messages specific to this probe. For information about generic Netcool/OMNIbus error messages, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide*.

Table 7. Error messages

Error	Description	Action
Cannot initialize the Orb	The probe found a problem during initialization of the ORB.	Ensure that the \$CLASSPATH environment variable contains the path to the IBM ORB . jar files.
Exception raised while creating ServiceMng object <i>exception</i>	The probe cannot retrieve the ServiceMng object.	Verify that the Naming Service is running and that the ServiceMng object is registered with the Naming Service.
Failed to get Comm Service <i>exception</i>	The probe cannot call the Comm Service.	Verify that the Naming Service is running and that the ServiceMng object is registered with the Naming Service.

Table 7. Error messages (continued)

Error	Description	Action
Failed to retrieve NotificationIRP object	The probe failed to get the NotificationIRP object specified by the NotificationObjectName property.	Verify that the value specified by the NotificationObjectName property is correct.
Failed to retrieve AlarmMng object	The probe failed to get the AlarmMng object specified by the AlarmObjectName property.	Verify that the value specified by the AlarmObjectName property is correct.
Problem while trying to connect to the corba objects <i>exception</i>	There was a problem connecting to the Alarm object or the Notification object.	Verify that the notification service is running.
Failed to ping notification service	The probe failed to ping the notification service or cannot get the Notification object.	Verify that the notification service is running.
There was a Runtime exception while pinging NotificationIRP <i>exception</i> Communication failure - lost connection to NotificationIRP	The probe encountered a runtime exception while pinging the notification service.	Verify that the notification service is running.
Detach Error	The probe failed to detach the subscription from the notification service.	You might need to restart the Naming Service.
NetcoolIRPManager: Failed to parse remainder_of_body. Type Any cannot be extracted using PropertySeqHelper	A problem occurred when the probe tried to parse an alarm.	Verify that the alarms are being generated in the correct alarm format.
Failed to perform resynch Could not perform resynch	The probe failed to get the alarm list or failed to iterate through resynch alarms.	Check that the connection to the host is still active.
Error while trying to attach the NetcoolIRPManager	The probe is not able to get a subscription to the notification service.	Verify that the ZTE N31 NMS server is running.
NetcoolIRPManager: Failed to iterate through resynch alarms	The probe encountered a problem while parsing the alarms retrieved during a resynchronization.	Verify that the ZTE N31 NMS server is running correctly. Verify that the Resynch property is set to true.

Table 7. Error messages (continued)

Error	Description	Action
Failed to acknowledge_alarms() Failed to Unacknowledge_alarms() Failed to clear_alarms Failed to comment_alarms	The probe could not acknowledge, unacknowledge, clear, or add a comment to one or more alarms.	Verify that the correct alarm ID was used with the CLI command. Verify that the correct command port is specified by the CommandPort property. Check the connection between the probe and the command port.
Failed to parse event completely	The probe encountered a problem while parsing the event	Verify that the ZTE N31 NMS server is running correctly.
Failed to send event	The probe could not forward an event to the ObjectServer.	Verify that the ObjectServer is running.
Name is null, cannot create Element	The probe could not forward an event to the ObjectServer.	Verify that the ObjectServer is running.
[Command Port] Failed to get CommandPortLimit property - using 10"); [Command Port] Failed to get property 'CommandPort'); [Command Port] Thread shutting down due to error!"); [Command Port] Failed to open listening socket, shutting down Thread!"); [Command Port] Error occurred : "+e); [Command Port] <"+host +"> Failed to get socket IO : "+io); [Command Port] <"+host +"> Failed to read command : "+e); [Command Port] <"+host +"> Failed to close command socket : "+se);	The probe could not retrieve the value of the CommandPortLimit property.	Verify that the correct command port is specified by the CommandPort property. Check the connection between the probe and the command port.

ProbeWatch messages

During normal operations, the probe generates ProbeWatch messages and sends them to the ObjectServer. These messages tell the ObjectServer how the probe is running.

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

Table 8. ProbeWatch messages		
ProbeWatch message	Description	Triggers or causes
START SYNCHRONIZATION	The probe is synchronizing the events.	The probe has started receiving alarms from the alarm list.
END SYNCHRONIZATION	The probe is closing the synchronization process.	The probe has finished receiving alarms from the alarm list.
Failed to connect to NotificationIRP: <i>reason</i> . Shutting down. Failed to Connect: Either, the server is not running, the naming service is down, or probe cannot reach remote Server Unexpected fatal error, failed to connect: <i>exception</i>	The probe could not connect to the target.	Check the exception message for indications of the cause of the problem.
Communication failure - lost connection to NotificationIRP	The probe lost the connection to the Notification object.	Verify that the notification service is running.

Running the probe

You run the probe from the command line.

Before you run the probe, update the default value of the **ServiceObjectStringRef** property with the host name and port number of the Naming Service server. This allows the ORB to access the string reference of the ServiceMng object.

The default value is:

```
corbaname:iiop:NamingServiceHost:NamingServicePort/StandardImplName/Naming/  
root#service
```

where *NamingServiceHost* is the IP address of the Naming Service host and *NamingServicePort* is the port number on which the Naming Service listens for connections.

To start the probe, use the following command:

```
$OMNIHOME/probes/nco_p_zten31_fixednms_corba
```

Running multiple probes

You can run multiple instances of the probe.

For each running instance, specify a different port to which the server listens using the **ORBLocalPort** property.

Running multiple probes in a failover configuration

If you have implemented a peer-to-peer failover configuration, both the master probe and the slave probe have their own Object Request Broker (ORB) local port.

Where you are running this configuration in conjunction with a firewall, add the receiving port of each probe to the firewall rules. The direction of the connection is from the target system to the master or slave probe.

You set the peer-to-peer failover mode in the properties files of the master and slave probes.

Include the following peer-to-peer settings in the master's probe properties file:

```
PeerHost      :      "slave_hostname"  
PeerPort      :      5555 # [communication port between master and slave probes]
```

Include the following peer-to-peer settings in the slave's probe properties file:

```
PeerHost      :      "master_hostname"  
PeerPort      :      5555 # [communication port between master and slave probes]
```

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.



SC27-2702-03

