IBM® Tivoli® Netcool/OMNIbus Probe for ZTE
M31 NetNumen WCDMA and U31 Wireless
(CORBA)
6.0

Reference Guide
November 7, 2014

IBM

Notice

Before using this information and the product it supports, read the information in Appendix A, "Notices and Trademarks," on page 27.

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 ZTE NetNumen M31 WCDMA (CORBA) documentation is provided in softcopy format only. To obtain the most recent version, visit the IBM® Tivoli® Knowledge Center:

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

Table 1. Document modification history		
Document version	**Publication date**	**Comments**
SC23-9571-00	July 25, 2008	First IBM publication.
SC23-9571-01	February 25, 2011	Support added for RAN (LTE) and CN (LTE). "Summary" on page 1 updated. Description for **FlushBufferInterval** added. Installation section replaced by "Installing probes" on page 2. "Firewall considerations" on page 3 added. The following properties were added to "Properties and command line options" on page 10: • **EncodingTo** • **FileTransferIrpName** • **InstanceID** • **ORBDebug** • **ORBDebugFileName** • **ORBDebugFilePath** • **TargetEMS** • **ValidateXML** • **XSDFile** • **XSDFtpCommand** • **XSDFtpPassword**

Table 1. Document modification history (continued)		
Document version	**Publication date**	**Comments**
SC23-9571-02	March 02, 2012	Information about operating system conventions added in "Conventions used in this guide" on page vi.
		Requirements and multicultural support information 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 10:
		• **ORBCharEncoding**
		• **ORBLocalHost**
		• **ORBWCharDefault**
SC23-9571-03	July 06, 2012	Probe target information updated in "Summary" on page 1.
		New information added to explain target EMS settings in "Setting the target EMS" on page 3.
		TargetEMS property updated in "Properties and command line options" on page 10.
		Error messages updated in "Error messages" on page 20.
		ProbeWatch messages updated in "ProbeWatch messages" on page 25.
SC23-9571-04	November 7, 2014	"Summary" on page 1 updated.

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 M31 NetNumen WCDMA and U31 Wireless (CORBA)

ZTE NetNumen U31 element management system (EMS) is an integrated network operation and maintenance system that controls different interconnected networks and performs integrated network fault management, fault location, and performance analysis of WCDMA, LTE, and GSM networks. It provides a uniform CORBA, northbound, fault management interface, and can be used to manage both fixed and mobile network equipment.

ZTE NetNumen U31 replaces (amongst other systems) ZTE NetNumen M31 RAN (Radio Access Network) and ZTE NetNumen M31 CN (Core Network). In this reference documentation, the ZTE U31, M31 RAN, and M31 CN are referred to collectively as the ZTE EMS.

The Probe for ZTE M31 NetNumen WCDMA and U31 Wireless (CORBA) collects alarms from the ZTE EMS using a Common Object Request Broker Architecture (CORBA) 3rd Generation Partnership Project (3GPP) interface.

The probe is described in 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 ZTE NetNumen M31 WCDMA (CORBA).

Table 3. Summary	
Probe targets	ZTE NetNumen M31-R (RAN and CN) V3.10
	ZTE NetNumen M31 (CN) V12.11.30
	ZTE NetNumen M31 (UMTS RAN) V12.10.032FP003 and V12.11.40B5
	ZTE NetNumen M31 (LTE RAN) V12.10.040j
	ZTE NetNumen U31 R06 V12.12.10 (which is another EMS R06)
	ZTE NetNumen U31 R07 V12.12.20 and V12.11.40.P01.B7
Probe executable name	`nco_p_zte_corba_wcdma`

Table 3. Summary (continued)	
Probe installation package	`omnibus-`*`arch`*`-probe-nco-p-zte-corba-wcdma-`*`version`*
Package version	6.0
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=swg21599279
Properties file	`$OMNIHOME/probes/`*`arch`*`/nco_p_zte_corba_wcdma.props`
Rules file	`$OMNIHOME/probes/`*`arch`*`/nco_p_zte_corba_wcdma.rules`
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
Remote connectivity	The probe can connect to a remote device using a 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 *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:

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.

Setting the target EMS

Use the **TargetEMS** property to specify which version of the ZTE EMS that you are running the probe against.

Different versions of the ZTE EMS use different CORBA interface definition language (IDL) files to specify the interfaces that CORBA objects present to the probe. Some of the IDL files used are specific implementations by individual communications companies. The value specified by the **TargetEMS** property tells the probe which set of IDL files are being used when it is communicating with a particular version of the ZTE EMS.

The following table lists the appropriate **TargetEMS** property setting for each supported ZTE EMS and its associated IDL files.

Table 4. Property settings for each supported ZTE EMS and IDL files		
TargetEMS property setting	Target ZTE EMS	IDL files
RAN	M31 UMTS RAN or LTE RAN	Hong Kong CSL IDL
CN	M31 CN	Hong Kong CSL IDL
M31_CN_UNICOM	M31 CN	China Unicom IDL
M31_RAN_UNICOM	M31 UMTS RAN	China Unicom IDL
U31	U31 R06 (only for China IDL) U31 R07	Default universal IDL or China Unicom IDL
U31_HKCSL	U31 R07	Hong Kong CSL IDL

Data acquisition

The probe gathers events from ZTE EMS using a 3GPP CORBA interface.

Data acquisition is further described in the following topics:

Connecting to the CORBA interface

The probe acts as an Integration Reference Point (IRP) Manager and connects to the ZTE EMS using a 3GPP CORBA interface. The probe connects to the interface using Interoperable Object Reference (IOR) files.

Before it begins retrieving events from the ZTE EMS, the probe acquires the `EntryPointIRP` CORBA object. It gets this from the local IOR file specified by the **EntryPointIrpFile** property.

After it has acquired the `EntryPointIRP` object, the probe acquires the `AlarmIRP` and `NotificationIRP` objects. It needs the `AlarmIRP` object to perform resynchronization and the `NotificationIRP` object to get real time alarms.

The probe uses the value specified by the **AlarmIrpName** property to locate the `AlarmIRP` object within the `EntryPointIRP` object. It uses the value specified by the **NotificationIrpName** property to locate the `NotificationIRP` object within the `EntryPointIRP` object.

Using FTP

If the Entry Point IRP file is on a remote system, the probe uses an FTP command to retrieve it. To enable this, you must specify the following value for the **EntryPointIrpFtpCommand** property:

```
ftp://user:password@host/path
```

Where *user* and *host* are the user name and host name of the Entry Point IRP server, *password* is specified by the **FtpPassword** property, and *path* is the location of the Entry Point IRP file as specified by the **EntryPointIrpFile** property.

Note : Encrypt the password using the `nco_g_crypt` utility supplied with Netcool/OMNIbus and use the encrypted string as the value of the **FtpPassword** property. For more information about encrypting passwords, see the *IBM Tivoli Netcool/OMNIbus Administration Guide* (SC14-7605).

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` point.

The probe then connects to the `NotificationIRP` point 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.

XML files

If the number of alarms in the ZTE EMS exceeds a maximum number (as configured in the EMS), the EMS puts all the active alarm data into an XML file. The probe then parses the XML file to retrieve the alarm data.

If the probe is running against ZTE NetNumen M31 CN, you can make the probe validate the XML file against an XML schema document (XSD) before processing the alarm data. If the probe is running against ZTE NetNumen M31 RAN, the probe parses the XML file without validating it first.

The properties related to XML validation can only be used when the value of the **TargetEMS** property is CN.

To enable XML validation, specify a value of `true` for the **ValidateXML** property and specify the location of the XML schema using the **XSDFile** property.

Using FTP

If the XML schema file is on a remote system, the probe can retrieve it over FTP. To enable this, you must specify the following value for the **XSDFtpCommand** property:

```
ftp://user:password@host/
```

Where *user* and *host* are the user name and host name of the server where the schema is located, and *password* is specified by the **XSDFtpPassword** property.

Note : Encrypt the password using the `nco_g_crypt` utility supplied with Netcool/OMNIbus and use the encrypted string as the value of the **XSDFtpPassword** property. For more information about encrypting passwords, see the *IBM Tivoli Netcool/OMNIbus Administration Guide* (SC14-7605).

Persistent subscriptions

The probe can maintain a subscription to the ZTE EMS across sessions.

To maintain a subscription across sessions, specify a value of true for the **PersistentSubscription** property.

If this property is set to false, the probe creates a new subscription to the ZTE EMS each time it makes a connection.

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.

Filtering notifications and alarms

You can use the `NotificationFilter` and `AlarmFilter` properties to limit the alarms and notifications that the probe acquires from the ZTE EMS.

When you use these properties, you must use the actual token names. For example, the token h represents the element NV_PERCEIVED_SEVERITY. So to specify that the probe is only sent notifications with a perceived severity of 3, set the `NotificationFilter` property to $h == 3.

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

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

You can use e and h as notification filter or alarm filter properties.

If the probe is acquiring events from ZTE LTE RAN, you can use c, d, e, f, g, h, l, m, n, s, y, z, jj, b, kk, a, k, ll, u, mm for alarm filtering, and c, d, e, b.time, h for notification filtering.

If the probe is acquiring events from ZTE LTE CN, you can use c, d, e, f, g ,h, k.time, l, m, n, s, y, z, b.time for alarm filtering, and c, d, e, b.time for notification filtering.

Note : To filter for b.time or b, the values must be in corba time and not real time.

The following table lists the token mappings that can be used with the `AlarmFilter` and `NotificationFilter` properties:

Table 5. Token mappings	
Element	**Token**
NV_NOTIFICATION_ID	a
NV_EVENT_TIME	b
NV_SYSTEM_DN	c
NV_MANAGED_OBJECT_CLASS	d
NV_MANAGED_OBJECT_INSTANCE	e

Table 5. Token mappings (continued)	
Element	**Token**
NV_ALARM_ID	f
NV_PROBABLE_CAUSE	g
NV_PERCEIVED_SEVERITY	h
NV_SPECIFIC_PROBLEM	i
NV_ADDITIONAL_TEXT	j
NV_ACK_TIME	k
NV_ACK_USER_ID	l
NV_ACK_SYSTEM_ID	m
NV_ACK_STATE	n
NV_COMMENTS	o
NV_BACKED_UP_STATUS	p
NV_BACK_UP_OBJECT	q
NV_THRESHOLD_INFO	r
NV_TREND_INDICATION	s
NV_STATE_CHANGE_DEFINITION	t
NV_MONITORED_ATTRIBUTES	u
NV_PROPOSED_REPAIR_ACTIONS	v
NV_CORRELATED_NOTIFICATIONS	w
NV_REASON	x
CLEAR_USER_ID	y
CLEAR_SYSTEM_ID	z
NV_ALARM_LIST_ALIGNMENT_REQUIREMENT	ff
NV_SERVICE_USER	gg
NV_SERVICE_PROVIDER	hh
NV_SECURITY_ALARM_DETECTOR	ii
NV_VENDOR_SPECIFIC_ALARM_TYPE	jj

Table 5. Token mappings (continued)	
Element	**Token**
NV_ALARM_RAISED_TIME	kk
NV_ALARM_CLEARED_TIME	ll
NV_ALARM_CHANGED_TIME	mm
NV_NOTIFICATION_CATEGORY_SET	zz
AI_VS_PERCEIVED_SEVERITY	ai_ps
AI_VS_ALARM_TYPE	ai_at
AI_VS_OTHER	ai_vs_other

Lookup table

The probe is supplied with a lookup table that contains details of the various types of alarms that ZTE EMS generates. You might need to update the rules file to include the path to the lookup table.

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

`$OMNIHOME/probes/includes/`

The following line in the rules file references the lookup file:

`include "../includes/Corba_zte_wcdma.lookup"`

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

Command line interface

The probe is supplied with a Command Line Interface (CLI). This interface allows you to execute commands using the probe 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.

Note : If there is a firewall between the probe and the ZTE EMS, 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 6. CLI commands	
Command	**Description**
exit/quit	Use this command to close the connection.
help	Use this command to display online help about the CLI.

Table 6. CLI commands (continued)	
Command	**Description**
resynch_all	Use this command to perform a full resynchronization with the 3GPP interface.

Resynchronization using the command line is only possible when the **Resynch** property is set to `true`. |
| **resynch_filter** *filter* | Use this command to perform partial resynchronization with the 3GPP interface.

This command uses the filter specified by the **AlarmFilter** property as a parameter. |
| **userid_acknowledge_alarm** *alarm_id user_id* | Use this command to acknowledge an alarm in the 3GPP interface by specifying the NV_ALARM_ID identifier of the alarm and the user identifier of the user acknowledging the alarm.

Before using this command, you must specify a value for the **AckSystemId** property. |
| **userid_unacknowledge_alarm** *alarm_id user_id* | Use this command to unacknowledge an alarm in the 3GPP interface by specifying the NV_ALARM_ID identifier of the alarm and the user identifier of the user acknowledging the alarm.

Before using this command, you must specify a value for the **AckSystemId** property. |
| **userid_clear_alarm** *alarm_id user_id* | 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 using this command, you must specify a value for the **ClearSystemId** property. |
| **userid_comment_alarm** *alarm_id user_id commentText* | 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 using 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 7. Properties and command line options

Property name	Command line option	Description
AckSystemId *string*	`-acksystemid` *string*	Use this property to specify the system identifier to use when acknowledging alarms using the CLI. The default is `Default`. **Note :** You must provide the system ID of the processing system instead of the default null value.
Agentheartbeat *integer*	`-agentheartbeat` *integer*	Use this property to specify the frequency (in seconds) with which the probe checks the status of the ZTE EMS server. The default is 300.
AlarmFilter *string*	`-alarmfilter` *string*	Use this property to specify the filter that the Alarm IRP uses to limit the alarms sent to the probe. The default is `" "`.
AlarmIrpName *string*	`-alarmirpname` *string*	Use this property to specify the name of the Alarm IRP Agent that the probe uses to resolve the AlarmIRP point in the Naming Service. The default is `ALARM IRP V1.0.0`.
ClearSystemId *string*	`-clearsystemid` *string*	Use this property to specify the system identifier to use when clearing alarms using the CLI. The default is `Default`.
CommandPort *integer*	`-commandport` *integer*	Use this property to specify the port through which you can use Telnet to send commands using the CLI. The default is 6970.
CommandPortLimit *integer*	`-commandportlimit` *integer*	Use this property to specify the maximum number of Telnet connections that the probe can make. The default is 10.
CommentSystemId *string*	`-commentsystemid` *string*	Use this property to specify the system identifier to use when adding comments to an alarm using the CLI. The default is `Default`.

Table 7. Properties and command line options (continued)

Property name	Command line option	Description
EncodingTo *string*	`-encodingto` *string*	Use this property to specify the encoding standard used by the probe. The default is `ISO-8859-1`. **Note :** When connecting to a ZTE EMS running in Chinese mode, specify the following value for this property: GB18030.
EntryPointIrpFile *string*	`-entrypointirpfile` *string*	Use this property to specify the location of the EntryPoint IRP file. The default is `""` .
EntryPointIrpFtp Command *string*	`-entrypointirpftpcommand` *string*	Use this property to specify the complete FTP command that the probe uses to access a remote Entry Point IRP file. The default is `""`.
FileTransferIrpName *string*	`-filetransferirpname` *string*	Use this property to specify the name that the probe uses to resolve the FileTransferIRP point in the Naming Service. The default is `FILETRANSFER IRP V1.0.0`. **Note :** The default value is for use with ZTE NetNumen M31 CN. Use the following value for ZTE NetNumen M31 RAN: `FileTransfer IRP V1.0`
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).
FtpPassword *string*	`-ftppassword` *string*	Use this property to specify the FTP password that the probe uses to access the Entry Point IRP file over FTP. The default is `""`. **Note :** Encrypt the password using the `nco_g_crypt` utility supplied with Netcool/OMNIbus.

Table 7. Properties and command line options (continued)

Property name	Command line option	Description
InstanceId *string*	-instanceid *string*	Use this property to specify a unique identifier for an instance of the probe. This property allows you to run multiple instances of the probe on the same machine. The default is `default`.
ManagerIdentifier *string*	-manageridentifier *string*	Use this property to specify the name of the Manager Identifier that receives all the IRPs. The default is `Netcool`.
NotificationCategories *string*	-notificationcategories *string*	Use this property to specify the notification categories to which the probe subscribes. To specify multiple categories, separate them using semicolons, in the following format: `category1;category2;category`*n* For example, use the following value to subscribe to alarms and heartbeat events: `ALARM IRP V1.0.0; CS IRP V1.0.0` The default is `" "` (the probe subscribes to all available notification categories).
NotificationFilter *string*	-notificationfilter *string*	Use this property to specify the filter that the Notification IRP uses to limit the notifications sent to the probe. This property uses the Extended Trader Constraint Language (Extended TCL) syntax defined in the OMG Notification Service specification. The default is `" "`.
NotificationIrpName *string*	-notificationirpname *string*	Use this property to specify the name that the probe uses to resolve the Notification IRP point in the Naming Service. The default is `NOTIFICATION IRP V1.0.0`.

Table 7. Properties and command line options (continued)		
Property name	**Command line option**	**Description**
ORBCharEncoding *string*	-orbcharencoding *string*	Use this property to specify the native character encoding set used by the Object Request Broker (ORB) for character data. This property takes the following values: • IS08859_1 • UTF8 The default is IS08859_1. **Note :** If the value of this property does not match the character encoding set used by the target system, the probe might not be able to perform resynchronization or get new notifications.
ORBDebug *string*	-orbdebug *string*	Use this property to specify whether the probe includes ORB messages in the debug file. This property takes the following values: false: The probe does not include ORB messages in the debug file. true: The probe includes ORB messages in the debug file. The default is false.
ORBDebugFileName *string*	-orbdebugfilename *string*	Use this property to specify the name of the ORB debug file. The default is orb.debug.
ORBDebugFilePath *string*	-orbdebugfilepath *string*	Use this property to specify the location of the ORB debug file. The default is ${OMNIHOME}/log/.
ORBLocalHost *string*	-orblocalhost *string*	Use this property to specify the local host used by the server-side ORB to place the server's host name into the IOR of a remote object. The default is "".
ORBLocalPort *integer*	-orblocalport *integer*	Use this property to specify the local port to which the Object Request Broker (ORB) listens. The default is 0 (the ORB selects an available port at random).

Table 7. Properties and command line options (continued)

Property name	Command line option	Description
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. This property takes the following values: • UTF16 • UCS2 The default is UTF16.
PersistentSubscription *string*	`-persistentsub` (This is equivalent to **PersistentSubscription** with a value of `true`.) `-nopersistentsub` (This is equivalent to **PersistentSubscription** with a value of `false`.)	Use this property to specify whether the probe maintains a persistent subscription to the ZTE EMS across sessions. This property takes the following values: `true`: The probe maintains a persistent subscription to the EMS. `false`: The probe does not maintain a persistent subscription to the EMS. The default is `false`.
Resynch *string*	`-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 existing active alarms from the ZTE EMS 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 *string*	`-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 EMS after losing the connection. This property takes the following values: `false`: The probe does not attempt to reconnect to the EMS. `true`: The probe attempts to reconnect to the EMS. The default is `false`.

Table 7. Properties and command line options (continued)		
Property name	**Command line option**	**Description**
TargetEMS *string*	`-targetems` *string*	Use this property to specify the target EMS to which the probe will connect to get alerts. This property takes the following values: • `RAN` • `CN` • `M31_CN_UNICOM` • `M31_RAN_UNICOM` • `U31` • `U31_HKCSL` The default is RAN.
Timeout *integer*	`-timeout` *integer*	Use this property to specify the time period (in seconds) that the probe waits for the next alarm before disconnecting. The default is `0` (which instructs the probe to maintain the connection indefinitely).
TimeTick *integer*	`-timetick` *integer*	Use this property to specify the lifetime (in minutes) of the Notification IRP session on the 3GPP server. The value of this property must match or exceed the default time tick setting on the ZTE EMS system. The default is `0` (the session remains open indefinitely).
ValidateXML *string*	`-validatexml` (This is equivalent to **ValidateXML** with a value of `true`.) `-novalidatexml` (This is equivalent to **ValidateXML** with a value of `false`.)	Use this property to specify whether the probe validates XML alarm data files received from ZTE NetNumen M31 CN. This property takes the following values: `false`: The probe does not validate the XML file. `true`: The probe validates the XML file. The default is `false`.
XSDFile *string*	`-xsdfile` *string*	Use this property to specify the location of the XML schema file (XSD) used to validate the XML alarm data file. The default is `" "`.

Table 7. Properties and command line options (continued)		
Property name	**Command line option**	**Description**
XSDFtpCommand *string*	`-xsdftpcommand` *string*	Use this property to specify the complete FTP command that the probe uses to access the XSD file. The default is `""`.
XSDFtpPassword *string*	`-xsdftppassword` *string*	Use this property to specify the password required to access the XSD file over FTP. The default is `""`. **Note :** Encrypt the password using the `nco_g_crypt` utility supplied with Netcool/OMNIbus.

Elements

The probe breaks event data down into tokens and parses them into elements. Elements are used to assign values to ObjectServer fields; the field values contain the event details in a form that the ObjectServer understands.

The following table describes the elements that the probe generates. Not all the elements described are generated for each event. The elements that the probe generates depends on the event type.

Table 8. Elements	
Element name	**Element description**
`AI_VS_ALARM_TYPE`	This element identifies the alarm type specific to the vendor.
`AI_VS_OTHER`	This element contains all the other details specific to the vendor.
`AI_VS_OTHER_ALARM_LOCATION_n`	This element provides the topological details specific to the vendor.
`AI_VS_OTHER_ALARM_SEQUENCE_ID`	This element contains the sequence number of the alert.
`AI_VS_OTHER_SERVER_ID`	This element indicates the other server specific to the vendor.
`AI_VS_OTHER_SYSTEM_TYPE`	This element shows the type of vendor system that generated the alarm.
`AI_VS_OTHER_TOPOLOGICAL_ID`	This element provides the other topological number specific to the vendor.
`AI_VS_PERCEIVED_SEVERITY`	This vendor specific element indicates the relative level of urgency for operator attention.

Table 8. Elements (continued)

Element name	Element description
CLEAR_SYSTEM_ID	This element identifies the system that cleared the alarm.
CLEAR_USER_ID	This element identifies the user that cleared the alarm.
NV_ACK_STATE	This element shows the acknowledgement state of the alarm.
NV_ACK_SYSTEM_ID	This element shows the system ID of the IRP Manager processing the notification.
NV_ACK_TIME	This element shows the time at which the user acknowledged the alarm.
NV_ACK_USER_ID	This element identifies the last user who changed the acknowledgement state.
NV_ADDITIONAL_TEXT	This element provides information about the network element from which the alarm originated.
NV_ALARM_CHANGED_TIME	This element contains the time when the alarm is changed.
NV_ALARM_CLEARED_TIME	This element contains the time when the alarm is cleared.
NV_ALARM_ID	This element shows the identification information of the alarm as it appears in the alarm list.
NV_ALARM_LIST_ALIGNMENT_REQUIREMENT	This element indicates whether the alarm list requires alignment.
NV_ALARM_RAISED_TIME	This element indicates the time when the alarm is raised.
NV_BACK_UP_OBJECT	This element shows the Distinguished Name (DN) of the backup object.
NV_BACKED_UP_STATUS	This element identifies whether the object has been backed up.
NV_COMMENTS	This element shows the comments about the alarms.
NV_COMMENTS_SYSTEM_ID	This element indicates the system used to raise comments.
NV_COMMENTS_TEXT	This element contains the text of the comment.

Table 8. Elements (continued)	
Element name	**Element description**
NV_COMMENTS_TIME	This element mentions the time when the comment is raised.
NV_COMMENTS_USER_ID	This element identifies the user that raised the comment.
NV_CORRELATED_NOTIFICATIONS	This element shows the correlated notifications.
NV_CORRELATED_NOTIFICATIONS_*notif_ID_Set*	This element shows the set of notifications to which this notification is considered to be correlated. This element is generated dynamically and its content is dependent on the IRP agent.
NV_CORRELATED_NOTIFICATIONS_SOURCE	This element shows the source of the notification set.
NV_EVENT_TIME	This element shows the time at which the event occurred.
NV_MANAGED_OBJECT_CLASS	This element shows the managed object class of the network resource.
NV_MANAGED_OBJECT_INSTANCE	This element shows the managed object instance of the network resource.
NV_MONITORED_ATTRIBUTE	This element shows the monitored attribute of the network resource.
NV_NOTIFICATION_ID	This element shows the identification information of the notification.
NV_NOTIFICATION_CATEGORY_SET	This element shows the category set information of the notification.
NV_PERCEIVED_SEVERITY	This element shows the relative level of urgency for operator attention.
NV_PROBABLE_CAUSE	This element provides further information about the probable cause of the alarm.
NV_PROPOSED_REPAIR_ACTIONS	This element shows the proposed repair actions associated with the notification.
NV_REASON	This element shows the reason that triggered the proposed repair action.
NV_SECURITY_ALARM_DETECTOR	This element shows the identifier of the detector of the security alarm.

Table 8. Elements *(continued)*	
Element name	**Element description**
NV_SERVICE_PROVIDER	This element shows the name of the service provider.
NV_SERVICE_USER	This element provides further information about the problem to which the notification relates.
NV_SPECIFIC_PROBLEM	This element provides further information about the problem to which the notification relates.
NV_STATE_CHANGE_DEFINITION	This element provides further information about the status change of the alarm.
NV_SYSTEM_DN	This element provides further information about the distinguished name (DN) used to identify the system.
NV_THRESHOLD_INFO	This element provides information about a threshold that has been crossed.
NV_TREND_INDICATION	This element indicates how an observed condition has changed.
NV_VENDOR_SPECIFIC_ALARM_TYPE	This element shows the alarm type specific to the vendor.

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 9. Error messages		
Error	**Description**	**Action**
Parser: ERROR when parsing an event *exception*	An event from the ZTE EMS could not be parsed. Some of the event elements might have been discarded.	Check the *exception* part of the message for more information about the cause of the error. If you cannot identify the cause of the error, contact IBM Software Support or the ZTE EMS administration team.
Parser: Stack Trace to stderr: *exception*	The probe encountered severe parser issues.	Probe cannot handle data sent from EMS and will provide more information by logging stack trace.
Parser: Failed to send event: *exception*	The probe could not forward the Event to the ObjectServer.	Check that the ObjectServer is running.

Table 9. Error messages (continued)

Error	Description	Action
`Parser: Exception raised in XML handling:` *`exception`*	The probe encountered a problem while parsing the XML file.	Verify that the following properties are correctly specified: **XSDFile**, **XSDFtpCommand**, and **XSDFtpPassword**.
`Parser: Failed to find XSD file '`*`xsd_file`*`'`	The probe could not retrieve the XSD file.	Verify that the following properties are correctly specified: **XSDFile**, **XSDFtpCommand**, and **XSDFtpPassword**.
`Parser: XML Validation failed`	The probe could not validate the XML configuration.	Verify that the XSD file being used is up to date with any EMS changes. Verify that the following properties are correctly specified: **XSDFile**, **XSDFtpCommand**, and **XSDFtpPassword**.
`Will not parse the XML`	The probe encountered a problem while parsing the XML file.	Verify that the following properties are correctly specified: **XSDFile**, **XSDFtpCommand**, and **XSDFtpPassword**.
`Parser: Not validating the incoming XML as the target EMS is:` *`TargetEMS_value`*	The probe is not validating the XML file.	The probe will only validate the XML file when the value specified by the **TargetEMS** property is CN.
`Error setting up SAX Parser to read Resynch XML file` *`exception`*	The probe cannot set up the SAX Parser.	Ensure that the $CLASSPATH environment variable is correctly specified. Verify that the following properties are correctly specified: **XSDFile**, **XSDFtpCommand**, and **XSDFtpPassword**.
`Parser: Failed to parse XML file` *`exception`*	The probe encountered a problem while parsing the XML file.	Check the *exception* part of the message for more information about the cause of this error. Verify that the following properties are correctly specified: **XSDFile**, **XSDFtpCommand**, and **XSDFtpPassword**.
`SaxZTEParser:Exception :` *`exception`*	The SAX parser encountered an exception.	Review the additional error messages and ProbeWatch messages for more information.
`SaxZTEParser: Failed to send event :` *`exception`*	The probe could not send an event to the ObjectServer.	Verify the connection to the ObjectServer and that the ObjectServer is running correctly.

Table 9. Error messages (continued)		
Error	**Description**	**Action**
`SAX Exception, XML Validation failed. Perhaps the XML is invalid` *exception*	The probe could not validate the XML configuration.	Verify that the following properties are correctly specified: **XSDFile**, **XSDFtpCommand**, and **XSDFtpPassword**.
`Error collecting and uncompressing XML file from target`	The probe failed to retrieve the XML file.	Verify that the following properties are correctly specified: **XSDFile**, **XSDFtpCommand**, and **XSDFtpPassword**.
`Probe unable to receive Resynchronization events`	The probe cannot receive resynchronization events.	Verify that the target EMS is running correctly. Verify that the **Resynch** property is correctly specified.
`Resynch events in XML file cannot be passed to ObjectServer`	The probe could not forward an event to the ObjectServer.	Verify that the ObjectServer is running.
`Error fetching Resynch property`	The probe could not retrieve the value specified by the **Resynch** property.	Verify that the **Resynch** property is correctly specified.
`Failed to perform resynch`	The probe failed to get the alarm list, or failed to iterate through resynchronization alarms.	Verify the connection to the target EMS and that the target EMS is running correctly.
`Login: Failed to find IOR file '`*iorFilePath*`'`	The probe failed to get the IOR file from the given FTP location.	Verify that the **EntryPointIrpFile**, **EntryPointIrpFtpCommand** and **FtpPassword** properties are correctly specified.
`Login: FtpPassword is incorrect length (ensure it has been encrypted using nco_g_crypt)`	The probe found that there is a problem with the encrypted password.	Verify that the **FtpPassword** property is correctly specified.
`Failed to find IOR file` *exception*	The probe failed to find the IOR file that stores subscription information for the previous session with the ZTE EMS.	Verify that the $OMNIHOME/var/*TargetEMSInstanceId*_NetcoolIRPManager.ior file exists and is readable.
`Name is null, cannot create Element`	The probe cannot get a name value for an event sent by the ZTE EMS, so the probe cannot process the event.	Check the log for related error messages with more information.

Table 9. Error messages (continued)

Error	Description	Action
No fault management file is identified in file listing from EMS	The probe failed to locate in the ZTE EMS the XML file that contains the current list of all active alarms.	Verify that an XML file for fault management was recently generated properly in the ZTE EMS.
Exception encountered. Listing available files *exception*	The probe is unable to list the files on the target EMS that contain resynchronization data.	Check that the EMS has created a file of event data.
Delete: no such file or directory: *filename*	The probe could not find the file or directory specified for deletion.	Check that you have specified the correct file or directory.
Delete: write protected: *filename*	The probe failed to overwrite a file or directory.	Check the permissions on the file or directory.
Delete: deletion failed: *filename*	The probe failed to delete the named file or directory.	Check the permissions on the file or directory.
PersistentSusbscription property incorrectly set - assuming false	An invalid value was specified for the **PersistentSubscription** property.	Verify that the value of the **PersistentSubscription** property. is either `true` or `false`. No other values are permitted.
Failed to unsubscribe from EMS *exception*	The probe failed to unsubscribe from ZTE EMS when shutting down. The subscription is still active in the ZTE EMS and this might cause problems in the EMS.	Check the *exception* part of the message for more information about the cause of this error.
Error sending probewatch	The probe failed to send a ProbeWatch message to the ObjectServer.	Verify the connection to the ObjectServer and that the ObjectServer is running correctly.
Error sending probe watch message on failed connection	The probe failed to get the Alarm IRP object that matches the value of the **AlarmIRP** property.	Verify the connection to the ObjectServer and that the ObjectServer is running correctly. Refer to your CORBA documentation for more information.
Error sending probe watch message on failed resynch	The probe failed to complete a resynchronization and could not send the related ProbeWatch message.	Verify the connection to the ObjectServer and that the ObjectServer is running correctly. Verify that the **Resynch** property is correctly specified.
Error sending probe watch message on starting synchronization	The probe encountered an error while sending the starting synchronization ProbeWatch message.	Verify the connection to the ObjectServer and that the ObjectServer is running correctly.

Table 9. Error messages (continued)		
Error	**Description**	**Action**
`Error sending probe watch message on successfully completed resynch`	The probe encountered an error while sending the completed resynchronization ProbeWatch message.	Verify the connection to the ObjectServer and that the ObjectServer is running correctly.
`Error sending probe watch message on failure to get subscription status`	The probe cannot retrieve the subscription status and cannot send the related ProbeWatch.	Verify the connection to the ObjectServer and that the ObjectServer is running correctly.
`Error sending end asynchronous resynch probe watch message`	The probe failed to send an asynchronous resynchronization ProbeWatch message.	Verify the connection to the ObjectServer and that the ObjectServer is running correctly.
`Error sending start asynchronous resynch probe watch message`	The probe failed to send an asynchronous resynchronization ProbeWatch.	Verify the connection to the ObjectServer and that the ObjectServer is running correctly.
`NetcoolIRPManager: Failed to get subscription status` *exception*	The probe cannot get the subscription status from the notification service.	Verify the connection to the target EMS and that the target EMS is running correctly.
`NetcoolIRPManager: Failed to Connect:` *exception* `NetcoolIRPManager: Either, the server is not running, the IOR is out of date, or probe cannot reach remote Server!`	The Notification IRP point from the target EMS is no longer responding.	The notification service is not available. Verify the connection to the target EMS and that the target EMS is running correctly.
`NetcoolIRPManager: Detach exception when handling old subscription` *exception*	The probe failed to detach from a previous subscription to the ZTE EMS before creating a new subscription.	Check the *exception* part of the message for more information about the cause of this error.
`NetcoolIRPManager: Failed to get alarm list` *exception*	The probe failed to retrieve the current list of active alarms from the ZTE EMS. The alarm list may not be synchronized between the ObjectServer and the ZTE EMS.	Check the *exception* part of the message for more information about the cause of this error.
`NetcoolIRPManager: Failed to iterate through resynch alarms` *exception*	The probe encountered a problem while parsing the alarms retrieved during a resynchronization.	Verify the connection to the target EMS and that the target EMS is running correctly. Verify that the **Resynch** property is correctly specified.

Table 9. Error messages (continued)

Error	Description	Action
`NetcoolIRPManager: Failed to acknowledge alarm(s) `*`exception`*	The probe failed to acknowledge an alarm in the ZTE EMS using the command port.	Verify that the command port commands were properly entered using the command line interface.
`NetcoolIRPManager: Failed to unacknowledge alarm(s) `*`exception`*	The probe failed to unacknowledge an alarm in the ZTE EMS using the command port.	Verify that the command port commands were properly entered using the command line interface.
`NetcoolIRPManager: Failed to comment alarm(s) `*`exception`*	The probe failed to add a comment to an alarm in the ZTE EMS using the command port.	Verify that the command port commands were properly entered using the command line interface.
`IO exception when reading IOR file: `*`exception`*	The probe failed to read the IOR file that stores subscription information for the previous session with the ZTE EMS.	Verify that the `$OMNIHOME/var/`*`TargetEMSInstanceId`*`_NetcoolIRPManager.ior` file exists and is readable.
`IO exception while reading old subscription id file=`*`subFileName exception`*	The probe failed to read the file that stores subscription information for the previous session with the ZTE EMS.	Verify that the *subFileName* file exists and is readable.
`Cannot initialize the Orb : `*`exception`*	The probe found a problem during initialization of the ORB.	Ensure that the `$CLASSPATH` environment variable contains the path to the IBM ORB `.jar` files.
`Failed to ping notification service`	The probe is not able to ping the notification service.	Verify the connection to the target EMS and that the target EMS is running 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 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 10. ProbeWatch messages

ProbeWatch message	Description	Trigger or cause
`Detaching subscription`	The probe is disconnecting from the target EMS.	The probe is restarting.
`Login: Failed to find IOR file '`*`iorFilePath`*`'`	The probe could not find the Entry Point IOR file on the local host.	Verify that the **EntryPointIrpFile** property is correctly specified and that the *iorFilePath* file is available.

Table 10. ProbeWatch messages (continued)		
ProbeWatch message	**Description**	**Trigger or cause**
`Failed to Connect: Either the server is not running, the IOR is out of date, or probe cannot reach remote Server`	The probe failed to connect to the Notification IRP on the ZTE EMS due to a CORBA OBJECT_NOT_EXIST exception.	Verify that the Notification IRP service is up and running on the ZTE EMS.
`Parser: Failed to find XSD file 'xsd_file'`	The probe could not find the named XSD file.	The values specified for the following properties might be incorrect: **XSDFile**, **XSDFtpCommand**, **XSDFtpPassword**.
`START SYNCHRONIZATION`	The probe is synchronizing the events.	The probe has started receiving alarms from the alarm list.
`END SYNCHRONIZATION`	The probe is ending the synchronization process.	The probe has finished receiving alarms from the alarm list.
`START ASYNCHRONOUS RESYNCH`	The probe is starting an asynchronous resynchronization.	The probe has started receiving alarms from the alarm list.
`END ASYNCHRONOUS RESYNCH`	The probe has completed an asynchronous resynchronization.	This probe has received the alarms from the alarm list.
`SYNCHRONIZATION ENDED UNSUCCESSFULLY. SEE PROBE ERROR LOG`	The probe failed to retrieve the current list of all active alarms from the ZTE EMS.	Check the probe log file for more information about this error.

Running the probe

Before running the probe for the first time, you must specify a minimum set of properties.

Before you run the probe, you must specify values for the following properties:

- **AlarmIrpName**
- **EntryPointIrpFile**
- **FileTransferIrpName**
- **NotificationIrpName**
- **TargetEMS**

If the Entry Point IRP file is on a remote system, you must also specify values for the following FTP properties:

- **EntryPointIrpFtpCommand**
- **FtpPassword**

For more information about these properties, see "Connecting to the CORBA interface" on page 4.

To start the probe, use the following command:

`$OMNIHOME/probes/nco_p_zte_corba_wcdma`

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.

SC23-9571-04