IBM® Tivoli® Netcool/OMNIbus Probe for ZTE
ZXONM E300 and U31 Wired (CORBA)
4.0

Reference Guide
July 6, 2012

IBM

Notice

Before using this information and the product it supports, read the information in Appendix A, "Notices and Trademarks," on page 21.

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 E300 (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**
SC27-2724-00	February 19, 2010	First IBM® publication.
SC27-2724-01	June 4, 2010	Summary table updated. Descriptions of the following properties were updated: • **ResyncProbableCauseFilter** • **ResyncSeverityFilter**
SC27-2724-02	March 31, 2011	Installation section replaced by "Installing probes" on page 2. "Firewall considerations" on page 3 added.
SC27-2724-03	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. The following new properties were added in "Properties and command line options" on page 6: • **ORBCharEncoding** • **ORBLocalHost** • **ORBWCharDefault**
SC27-2724-04	July 6, 2012	Probe target system information updated in "Summary" on page 1:. Additional property descriptions added to "Properties and command line options" on page 6:. Additional element descriptions added to "Elements" on page 13.

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 ZXONM E300 and U31 Wired (CORBA)

The ZTE E300, ZTE T3, and ZTE U31 R10 systems control different interconnected networks and perform integrated network fault management, fault location, and performance analysis. Their centralized reporting function provides multiple views of the networks and helps to compare their performance.

The Probe for ZTE ZXONM E300 and U31 Wired (CORBA) collects alarms from the ZTE E300, ZTE T3, and ZTE U31 systems using a Common Object Request Broker Architecture (CORBA) interface.

This guide contains the following sections:

Summary

Each probe works in a different way to acquire event data from its source, and therefore has specific features, default values, and changeable properties. Use this summary information to learn about this probe.

The following table provides a summary of the Probe for ZTE E300 (CORBA).

Table 3. Summary	
Probe target	U31 R10 V12.12.10 U31 R22 V12.12.20 E300 E300 V3.19 R2P05 T31/T37 T31 V1.20 (T37 V1.20 R1P01)
Probe executable name	`nco_p_zte_e300`
Probe installation package	`omnibus-`*`arch`*`-probe-nco-p-zte-e300-`*`version`*
Package version	4.0
Probe supported on	For details of supported operating systems, see the following Release Notice on the IBM Software Support website: https://www-304.ibm.com/support/docview.wss?uid=swg21599278
Properties file	`$OMNIHOME/probes/`*`arch`*`/zte_e300.props`

Table 3. Summary (continued)	
Rules file	`$OMNIHOME/probes/`*arch*`/zte_e300.rules`
Requirements	A currently supported version of IBM Tivoli Netcool/OMNIbus. 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	Not 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 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 system using a Common Object Request Broker Architecture (CORBA) interface.

Data acquisition is described in the following topics:

- "Connecting to the CORBA interface" on page 3
- "Retrieving alarms and notifications" on page 4
- "Command line interface" on page 4
- "Peer-to-peer failover functionality" on page 6

Connecting to the CORBA interface

The probe connects to the CORBA interface using Interoperable Object Reference (IOR) files.

The probe acquires the object reference of the `EmsSessionFactory_I` CORBA object from the IOR file specified by the **ObjectIORFile** property. If you did not specify a value for the **ObjectIORFile** property, the probe gets the object reference of the `EmsSessionFactory_I` object from the Naming Service, using the value specified by the **NamingContextPath** property.

The probe can locate the Naming Service in one of two ways:

1. By using the host name and port number of the Naming Service host specified by the **ORBInitialHost** and **ORBInitialPort** properties.

2. By using the IOR file specified by the **NSIORFile** property.

The Naming Service uses the value specified by the **NamingContextPath** property to locate the EmsSessionFactory_I object.

After the probe has acquired the reference to the EmsSessionFactory_I object, it logs into the ZTE system using the user name and password specified by the **Username** and **Password** properties. It then creates an EMS Session and queries the Subscriber and EMS manager objects. The Subscriber manager object is used to subscribe to realtime event notifications and the EMS manager object is used to perform resynchronizations.

Retrieving alarms and notifications

If the **Resync** property is set to true, the probe initially receives a list of all active alarms from the EMS manager object. The probe then connects to the Subscriber object and uses the CORBA notification push model to receive new alarms from the server as they are generated. If the **Resync** property is set to false, the probe only receives new alarms.

You can use the **ResyncProbableCauseFilter** and **ResyncSeverityFilter** properties to filter the alarms that the probe receives during resynchronization. Use the **ResyncProbableCauseFilter** property to filter alarms by their probable cause and the **ResyncSeverityFilter** property to filter alarms according to their level of severity.

Command line interface

The probe is supplied with a Command Line Interface (CLI) that enables you to issue commands to the probe using Telnet.

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.

Use the **CommandPortLimit** property to specify how many CLI connections can be open at the same time.

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

Table 4. CLI commands	
Command	**Description**
ackAlarm *<alarmId>* *<emsId>* *<managedElementId>* *<username>* **Note:** If connecting to the ZTE U31 R10 system, the emsId and managedElementId values are optional. You can specify any value to proceed with acknowledging the alarm.	Use this command to acknowledge an alarm, where: • *<alarmId>* is the identifier stored in the alarm's emsAlarmId field. • *<emsId>* is the identifier stored in the alarm's EMS field. • *<managedElementId>* is the identifier stored in the alarm's ManagedElement field. • *<username>* is the user name of the user acknowledging the alarm (the default is root).

Table 4. CLI commands (continued)

Command	Description
clearAlarm *<alarmId> <emsId> <managedElementId> <username>* **Note:** If connecting to the ZTE U31 R10 system, the emsId and managedElementId values are optional. You can specify any value to proceed with clearing the alarm.	Use this command to clear an alarm, where: • *<alarmId>* is the identifier stored in the alarm's emsAlarmId field. • *<emsId>* is the identifier stored in the alarm's EMS field. • *<managedElementId>* is the identifier stored in the alarm's ManagedElement field. • *<username>* is the user name of the user acknowledging the alarm (the default is root).
exit/quit	Use this command to close the connection.
help	Use this command to display online help about the CLI.
getEventQueueSize	Use this command to display the current size of the internal event queue.
name	Use this command to display the name of the probe.
resync	Use this command to perform a resynchronization using the values specified by the **ResyncSeverityFilter** and **ResyncProbableCauseFilter** properties.
resyncFilter *filter*	Use this command to perform a resynchronization using a custom filter. Custom filters take the following format: **resyncFilter** sev=*severity1*;*severity2*; pbCause=*pbCause1*;*pbCause2*; To perform a full resynchronization, use the following command: **resyncFilter** sev= pbCause= **Note:** This command does not accept spaces in the sev and pbCause arguments. If you want to use a filter that contains spaces, you must use the **ResyncSeverityFilter** and **ResyncProbableCauseFilter** properties.
shutdown	Use this command to shutdown the probe.
version	Use this command to print 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 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 5. Properties and command line options		
Property name	**Command line option**	**Description**
AgentHeartBeat *integer*	`-agentheartbeat` *integer*	Use this property to specify the frequency (in seconds) with which the probe checks the status of the host server. The default is 60.

Property name	Command line option	Description
CommandPort *integer*	`-commandport` *integer*	Use this property to specify a port through which you can send commands to the probe using the probe's CLI. The default is 6970.
CommandPortLimit *integer*	`-commandportlimit` *integer*	Use this property to specify the maximum number of CLI connections that you can have open at one time. The default is 10.
DataBackupFile *string*	`-databackupfile` *string*	Use this property to specify the location of the probe's data backup file. The default is `$OMNIHOME/var/zte_e300.bak`.

EncodingTo *string*	`-encodingto` *string*	Use this property to specify the character set used to display data received from the ZTE system.
		
		The default is `ISO-8859-1` (Latin-1).
		
		Note:
		
		When the ZTE system is using Chinese language settings, you must specify the following value for this property: GB18030. The probe creates an additional token called `VendorProbableCause_GB18030` to contain the original value.
		
		When the ZTE T3 or U31 system is using Chinese language settings, you must specify the following value for this property: GBK. This will enable the tokens Description and vendorProbablecause to displays Chinese characters. You must also have set the environment for OMNIbus and the probe to be GBK as well:
		
		`export LANG=zh_CN.GBK`
		
		`export LC_ALL=zh_CN.GBK`

Table 5. Properties and command line options (continued)

Table 5. Properties and command line options (continued)

Property name	Command line option	Description
Inactivity *integer*	`-inactivity` *integer*	Use this property to specify the length of time (in seconds) that the probe allows the port to be inactive before disconnecting. The default is 0 (which instructs the probe to not disconnect during periods of inactivity).
NamingContextPath *string*	`-nspath` *string*	Use this property to specify the name used to register the object in the Naming Service. The default is `""`.
NotificationFilter *string*	`-notificationfilter` *string*	Use this property to specify the filter to use when subscribing for notifications. For example: `$type_name == 'NT_ALARM'` The default is `""`.
NSIORFile *string*	`-nsior` *string*	Use this property to specify the location of the IOR file for the root context of the Naming Service. The default is `""`.
ObjectIORFile *string*	`-objectiorfile` *string*	Use this property to specify the location of the IOR file used by the probe to acquire the object reference. The default is `""`.
ORBCharEncoding *string*	`-orbcharencoding` *string*	Use this property to specify the native character encoding set used by the Object Request Broker (ORB) for character data. The default is UTF8.
ORBDebug *string*	`-orbdebug` *string*	Use this property to specify whether the probe writes ORB messages to a debug log file. This property takes the following values: `false`: The probe does not write ORB messages to a debug file. `true`: The probe writes ORB messages to a debug file. The default is `false`.

Table 5. Properties and command line options (continued)		
Property name	**Command line option**	**Description**
ORBDebugFile *string*	`-orbdebugfile` *string*	Use this property to specify the location of the ORB debug file. The default is `$OMNIHOME/log/ orb.debug`.
ORBInitialHost *string*	`-orbinitialhost` *string*	Use this property to specify the host name of the Naming Service server. The default is `" "`.
ORBInitialPort *integer*	`-orbinitialport` *integer*	Use this property to specify the port number through which to connect to the Naming Service host. The default is 1570.
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 for connections from the probe. The default is 0 (the ORB selects an available port at random).
ORBWCharDefault *string*	`orbwchardefault` *string*	Use this property to specify what wide character (wchar) set the IBM ORB uses when communicating with other ORBs that do not publish a wchar set. The default is UTF16
Password *string*	`-password` *string*	Use this property to specify the password of the user account used to connect to the ZTE system. The default is `" "`.

Table 5. Properties and command line options (continued)		
Property name	**Command line option**	**Description**
Resync *string*	`-resync` *string*	Use this property to specify whether the probe requests all active alarms from the host server on startup. This property takes the following values:
		false: The probe does not request resynchronization on startup.
		true: The probe requests resynchronization on startup.
		The default is `true`.
ResyncInterval *integer*	`-resyncinterval` *integer*	Use this property to specify the interval (in seconds) at which the probe makes successive resynchronization requests.
		The default is `0` (which instructs the probe to not make successive resynchronization requests).
ResyncBatchSize *integer*	`-resyncbatchsize` *integer*	Use this property to specify the maximum number of alarms retrieved in each batch of resynchronization alarms.
		The default is `100`.

Table 5. Properties and command line options (continued)		
Property name	**Command line option**	**Description**
ResyncProbableCause Filter *string*	`-resyncpbcausefilter` *string*	This property works differently depending on whether the probe is connecting to a ZTE E300, ZTE T3, or ZTE U31 R10 system. For the ZTE E300 system, use this property to specify a list of probable causes that the probe **excludes** when resynchronizing with the CORBA interface. For the ZTE T3 system, use this property to specify a list of probable causes that the probe **includes** when resynchronizing with the CORBA interface. For the ZTE U31 R10 system, use this property to specify a list of probable causes that the probe **includes** when resynchronizing with the CORBA interface. For each system, separate each cause in the list with a semicolon when specifying the value of the property. For example: `pbCause1;pbCause2;pbCause3` The default is `" "`.

Table 5. Properties and command line options (continued)

Property name	Command line option	Description
ResyncSeverityFilter *string*	`-resyncseverityfilter` *string*	This property works differently depending on whether the probe is connecting to a ZTE E300, ZTE T3, or ZTE U31 R10 system.
		For the ZTE E300 system, use this property to specify a list of severities that the probe **excludes** when resynchronizing with the CORBA interface.
		For the ZTE T3 system, use this property to specify the list of severities that the probe **includes** when resynchronizing with the CORBA interface.
		For the ZTE U31 R10 system, use this property to specify a list of severities that the probe **includes** when resynchronizing with the CORBA interface.
		For each system, separate each severity in the list with a semicolon when specifying the value of the property. For example, to receive only critical and major events, specify the following value:
		`PS_CRITICAL;PS_MAJOR`
		This property takes any combination of the following severities as its value:
		• PS_INDETERMINATE
		• PS_CRITICAL
		• PS_MAJOR
		• PS_MINOR
		• PS_WARNING
		• PS_CLEARED
		The default is `""`.
RetryMaxTime *integer*	`-retrymaxtime` *integer*	Use this property to specify the maximum length of time (in seconds) that the probe makes reconnection attempts to the target system. If you set this property to 0, the probe makes reconnection attempts indefinitely.
		The default is 9000.

Table 5. Properties and command line options (continued)		
Property name	**Command line option**	**Description**
RetryConstantWait *integer*	`-retryconstantwait` *integer*	Use this property to specify the length of time (in seconds) that the probe waits between successive connection attempts to the target system. The default is 0 (which instructs the probe to use an exponentially increasing period between successive connection attempts, for example, the probe will wait for 1 second, then 2 seconds, then 4 seconds, and so forth).
Username *string*	`-username` *string*	Use this property to specify the user name of the user account used to connect to the ZTE system. The default is " "

Elements

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

The following table describes the elements that the Probe for ZTE E300 (CORBA) generates. Not all the elements described are generated for each event; the elements that the probe generates depends upon the event type.

Table 6. Elements	
Element name	**Element description**
`$ackInfo`	This element contains additional information that was included with the alarm when it was acknowledged.
`$ackTime`	This element displays the time at which the alarm was acknowledged.
`$ackUser`	This element indicates the user who acknowledged the alarm.
`$AID`	This element displays the alarm identifier.
`$alarmKey`	This element contains the alarm identifier.
`$AlarmSource`	This element indicates the source of the alarm.
`$AlarmStatus`	This element indicates the status of the alarm.
`$alarmType`	This element indicates the alarm type.

Table 6. Elements (continued)

Element name	Element description
$ClearStatus	This element indicates the clear status of the alarm.
$clearTime	This element displays the time at which the alarm was cleared.
$CTP	This element displays the CTP code of the device.
$ClockSource	This element indicates the clock source.
$CorrelatedAlarmIds	This element contains the IDs of the alarms that have been correlated with this alarm.
$domain_name	This element displays the domain name on which the E300 server is running.
$Description	This element provides a description of the alarm. If the ZTE T3 or U31 system is running with Chinese settings, this element will contain Chinese characters.
$DiagnoseInfo	This element contains information about the event that caused the alarm.
$EMS	This element displays the name of the element management system (EMS).
$emsAlarmId	This element displays the identifier of the alarm used by the EMS.
$emsTime	This element displays the time the alarm was generated by the EMS.
$Equipment	This element displays the name of the equipment being reported.
$EquipmentHolder	This element displays the name of the equipment holder.
$event_name	This element displays a description of the event being reported.
$event_type	This element displays the type of the event.
$FTP	This element displays the FTP code of the device.
$ifRevertive	This element indicates whether the status can be reverted.
$layerRate	This element displays the layer rate.
$ManagedElement	This element displays the name of the network element being monitored.

Element name	Element description
$nativeEMSName	This element displays the name of the network element as used by the EMS.
$NotifyFilter	This element displays the notification filter.
$objectFilterName	This element displays the name of the filter used to select the objects that the probe receives.
$perceivedSeverity	This element displays the perceived severity of the alarm.
$PGP	This element displays the PGP code of the device.
$pmValue	This element displays the pulse modulation value.
$probableCause	This element displays the probable cause of the alarm.
$PTP	This element displays the PTP code of the device.
$raiseTime	This element displays the time at which the alarm was raised.
$sendTime	This element displays the time at which the alarm was sent.
$serviceAffecting	This element displays the name of the service affected.
$switchPosition	This element displays the position of the switch.
$switchReason	This element indicates the reason that the switch failed.
$switchStatus	This element displays the status of the switch.
$systemName	This element displays the name of the system affected.
$VendorProbableCause	This element is generated when the probe is run against the E300 system. It displays the probable cause of the fault as defined by the vendor.
$vendorProbableCause	This element is generated when the probe is run against T3 and U31 systems. It displays the probable cause of the fault as defined by the vendor. If the ZTE T3 or U31 system is running with Chinese settings, this element will contain Chinese characters.

Table 6. Elements (continued)

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
`Failed to create new event :`	The event manager was unable to create a new event.	Check that the ZTE server is running correctly.
`Failed to extract values from the remainder_of_body`	There was an error while extracting data from one of the fields of the CORBA event.	Check that there is not an IDL type mismatch.
`Failed to send event :`	The event manager could not send the event to the ObjectServer.	Check that the ObjectServer is running. Check that the network connection is working correctly.
`Name is null, cannot create element`	The name of the attribute being parsed is not recognized	Check that there is not an IDL type mismatch.
`Cannot parse type[`	The type of the attribute being parsed is not recognized	Check that there is not an IDL type mismatch.
`Failed to parse element :`	The element 'name' could not be parsed.	Check that there is not an IDL type mismatch.
`Failed to get object reference from IOR file :`	The probe failed to obtain a reference on the EmsSessionFactory object by reading its IOR file.	Check that the IOR file exists and it is up-to-date. Check that the network connection between the machine where the probe is running and the machine where the E300 is located is working correctly.
`Failed to connect to Naming Service :`	The probe could not connect to the Naming Service on the host and the port specified by the **ORBInitialHost** and **ORBInitialPort** properties.	Check that the Naming Service is running. Check that the network connection is working correctly.
`Failed to get object from IOR file :`	The probe failed to get a reference on the Naming Service by reading its IOR file specified in the **NSIORFile** property.	Check that the Naming Service is running. Check that the network connection is working correctly.
`Failed to resolve Naming Context`	The probe is connected to the Naming Service but could not obtain a reference to the object whose name is specified in the **NamingContextPath** property.	Check that the object name is correct and it's currently available.

Table 7. Error messages (continued)

Error	Description	Action
Failed to narrow object reference :	The probe got a reference on a CORBA object but fails to narrow it into a EmsSessionFactory object reference.	Check the IOR files and check the NamingContextPath is correct and up-to-date.
Failed to convert IOR to object :	The probe failed to convert an object reference into an IOR	Check that the ZTE server is running correctly.
Failed to find file, please check file exists!	The probe is unable to locate a file.	Check that the file exists.
The CORBA service is not running on the expected Host/ Port/IOR :	The Naming Service is not running on the expected Host/Port/IOR.	Check the Naming Service is running and then check that all the associated properties have been defined correctly.
Exception raised when locating server : "Check Naming Service and CORBA interface is running!	The object name specified by the **NamingContextPath** property cannot be found.	Check the object is available and the specified name is correct.
Failed to generate and process Probewatch event for the event loss being cleared.	The ProbeWatchManager failed to create a ProbeWatch message.	Check the ObjectServer is running.
Failed generate and process Probewatch event for the event loss occurring.	The ProbeWatchManager failed to create a ProbeWatch message.	Check the ObjectServer is running.
ZTEE300Probe: error while checking subscription status :	The probe received an exception while calling getSubscriptionStatus on the EMSSubscriptionManager.	Check that the ZTE server is running correctly.
Failed to create CorbaUtil object :	The probe could not create the internal object.	Ensure that the $CLASSPATH environment variable contains the path to the IBM ORB .jar files.
Failed to get object reference :	The probe could not get a reference on the EmsSessionFactory object.	Check that the associated details are correct.
Please check Username and Password""Unable to get EMS session :	The probe received an authentication error while trying to retrieve an EmsSession object.	Check that the **Username** and **Password** properties have been specified correctly.
Unable to get EMS session :	The probe received an exception while trying to retrieve an EmsSession object.	Check that the ZTE server is running correctly.

Table 7. Error messages (continued)

Error	Description	Action
`Failed to get EMS Manager reference :`	The probe received an exception while trying to retrieve the EMSManager reference from the EmsSession object.	Check that the ZTE server is running correctly.
`Failed to narrow manager reference :`	The object returned by the EmsSession is not an EMSManager.	Check that the ZTE server is running correctly.
`ZTEE300Probe: Failed to get object reference :`	The probe received an exception while trying to retrieve the EMSManager reference from the EmsSession object.	Check that the ZTE server is running correctly.
`ZTEE300Probe: problem while unsubscribing :`	The probe received an exception was while calling unsubscribe with the probe's subscriptionId.	Check that the ZTE server is running correctly.
`Failed to parse alarm :`	The probe received an exception while parsing an alarm.	Check that there is not an IDL type mismatch.
`Failed to get resynch alarm list :`	The probe received an exception while calling getAllActiveAlarms.	Check the **ResyncProbableCauseFilter** and **ResyncBatchSize** filters have the correct/supported values. Check that the ZTE server is running correctly.
`Failed to retrieve next batch of resync alarms :`	The probe failed to retrieve the next batch of alarms.	Check that the ZTE server is running correctly.
`Failed to perform resync :`	The probe received an exception during resynchronization.	Check that the ZTE server is running correctly.
`ZTEE300Probe: Failed to get 'Subscriber' interface reference :`	The probe failed to retrieve a reference on the subscriber manager.	Check that the ZTE server is running correctly.
`ZTEE300Probe: Failed to narrow manager reference :`	The probe retrieved an incorrect object type.	Check that the ZTE server is running correctly.
`ZTEE300Probe: failed to subscribe :`	The probe could not subscribe for real-time notifications.	Check that the ZTE server 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 8. ProbeWatch messages

ProbeWatch message	Description	Triggers/causes
`START SYNCHRONIZATION`	Resynchronization is in progress.	The probe started up, or the period specified by the **ResyncInterval** property has elapsed since the last resynchronization.
`END SYNCHRONIZATION`	The resynchronization has completed.	The probe received all events that matched the filtering criteria.
`ClientSession event loss cleared.`	An event loss was cleared.	ZTE E300 indicated that an event loss has now been cleared.
`ClientSession event loss occurred.`	An event loss occurred.	ZTE E300 indicated that the events have been lost.

Running the probe

You run the probe from the command line.

To start the probe, use the following command:

`$OMNIHOME/probes/nco_p_zte_e300`

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.

SC27-2724-04