

Netcool/OMNIbus TL1 Probe
3.0

Reference Guide
December 11, 2014



Note

Before using this information and the product it supports, read the information in [Appendix A, “Notices and Trademarks,” on page 29.](#)

Edition notice

This edition (SC14-7631-02) applies to version 3.0 of IBM Tivoli Netcool/OMNIbus TL1 Probe and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces (SC14-7631-01).

© **Copyright International Business Machines Corporation 2006, 2014.**

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

Contents

About this guide.....	v
Document control page.....	v
Conventions used in this guide.....	vi
 Chapter 1. TL1 Probe.....	1
Summary.....	1
Installing probes.....	2
Migrating from the TL1 TSM.....	2
Data acquisition.....	3
TL1 protocol.....	4
Issuing commands.....	4
TL1 command structure.....	4
Message structure and expected tokens.....	6
Multiple events per message.....	6
Peer-to-peer failover functionality.....	7
Connecting to single or multiple devices.....	8
Using chat in scripts and chat out scripts.....	9
Alerting TL1 devices to the probe connection.....	9
Resynchronizing with the TL1 device.....	9
Heartbeating.....	9
Specifying devices using a hosts file.....	10
Tracking the login status of the network elements on a TL1 device.....	12
Read timeout.....	12
Backoff Strategy.....	12
Data Stream Capture.....	13
Command line interface.....	13
Issuing commands to the probe.....	14
Properties and command line options.....	15
Escape Codes.....	20
Running the probe	21
Elements.....	21
Static elements.....	21
ProbeWatch Message elements.....	22
Dynamic elements.....	22
Error messages.....	23
ProbeWatch messages.....	26
Known issues.....	28
 Appendix A. Notices and Trademarks.....	29
Notices.....	29
Trademarks.....	30

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 TL1 Probe 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
SC14-7631-00	October 29, 2010	First IBM publication.
SC14-7631-01	November 30, 2012	“Specifying devices using a hosts file” on page 10 updated. The following command descriptions are updated in “Properties and command line options” on page 15 • ChatinString • HostsFile
SC14-7631-02	December 11, 2014	“Summary” on page 1 updated. Details about reducing event traffic when running TL1 probes in a peer-to-peer failover configuration using the SlaveMode property added to “Peer-to-peer failover functionality” on page 7. Information about the supported hosts file formats updated in “Format for hosts file when specifying multiple TL1 devices on a single host” on page 11 “Tracking the login status of the network elements on a TL1 device” on page 12 added. Details about using the StreamCaptureFileAppend property to specify that the probe appends the stream capture file rather than overwriting it added to “Data Stream Capture” on page 13. Descriptions of the SlaveMode property and the StreamCaptureFileAppend property added to “Properties and command line options” on page 15.

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. TL1 Probe

Transaction Language 1 (TL1) is a subset of the input and output messages contained in the International Telecommunications Union (ITU) Man-Machine Language (MML) command-line interface. TL1 provides a standard set of messages that can be used for communicating between operating systems and network elements (NEs), and between personnel and NEs. Functions such as surveillance, memory administration, and test access use short messages to cause specific actions. Transactions include the initiation and execution of these messages. Because these messages are standardized, they ensure compatible connections between any equipment designed to meet the TL1 standards.

The Transaction Language 1 Telco Service Manager (TL1 Probe) gathers fault and event data from all TL1-managed elements in a network and forwards the data as alerts to the Netcool/OMNIbus ObjectServer for correlation, consolidation, and acknowledgement by network operators.

This guide contains the following sections:

- [“Summary” on page 1](#)
- [“Installing probes” on page 2](#)
- [“Data acquisition” on page 3](#)
- [“Properties and command line options” on page 15](#)
- [“Running the probe ” on page 21](#)
- [“Elements” on page 21](#)
- [“Error messages” on page 23](#)
- [“ProbeWatch messages” on page 26](#)

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 TL1 Probe.

<i>Table 3. Summary</i>	
Probe target	TL1 Protocol
Probe executable name	nco_t_tl1
Package version	3.0
Probe supported on	For details of supported operating systems, see the following Release Notice on the IBM Software Support website: http://www-01.ibm.com/support/docview.wss?uid=swg21450079
Properties file	\$OMNIHOME/tsm/arch/tl1.props
Rules file	\$OMNIHOME/tsm/arch/tl1.rules
Requirements	A currently supported version of IBM Tivoli Netcool/OMNIbus.
Connection method	TCP/IP
Multicultural support	Not Available

Table 3. Summary (continued)	
Peer-to-peer failover functionality	Available
IP environment	IPv4 and IPv6 Note : The probe is supported on IPv6 when running on IBM Tivoli Netcool/OMNIBus V7.3.0, 7.3.1 and 7.4.0 on all UNIX and Linux operating systems.
Federal Information Processing Standards (FIPS)	IBM Tivoli Netcool/OMNIBus uses the FIPS 140-2 approved cryptographic provider: IBM Crypto for C (ICC) certificate 384 for cryptography. This certificate is listed on the NIST website at http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2004.htm . For details about configuring Netcool/OMNIBus for FIPS 140-2 mode, see the <i>IBM Tivoli Netcool/OMNIBus Installation and Deployment Guide</i> .

Installing probes

All probes are installed in a similar way. The process involves downloading the appropriate installation package for your operating system, installing the appropriate files for the version of Netcool/OMNIBus that you are running, and configuring the probe to suit your environment.

The installation process consists of the following steps:

1. Downloading the installation package for the probe from the Passport Advantage Online website.

Each probe has a single installation package for each operating system supported. For details about how to locate and download the installation package for your operating system, visit the following page on the IBM Tivoli Knowledge Center:

http://www-01.ibm.com/support/knowledgecenter/SSSHTQ/omnibus/probes/all_probes/wip/reference/install_download_intro.html

2. Installing the probe using the installation package.

The installation package contains the appropriate files for all supported versions of Netcool/OMNIBus. For details about how to install the probe to run with your version of Netcool/OMNIBus, visit the following page on the IBM Tivoli Knowledge Center:

http://www-01.ibm.com/support/knowledgecenter/SSSHTQ/omnibus/probes/all_probes/wip/reference/install_install_intro.html

3. Configuring the probe.

This guide contains details of the essential configuration required to run this probe. It combines topics that are common to all probes and topics that are peculiar to this probe. For details about additional configuration that is common to all probes, see the *IBM Tivoli Netcool/OMNIBus Probe and Gateway Guide*.

Migrating from the TL1 TSM

Migrating from the TL1 TSM entails converting the SERVERLOG information used by the TL1 TSM into a format that can be copied into the hosts file used by the TL1 Probe to connect to your TL1 devices.

To migrate from the TL1 TSM to the TL1 Probe, use the following steps:

1. Backup the MySQL tsm database into a temporary file using the following command:

```
../tl1tsm/mysql/bin>./mysqldump tsm > tsm.sql
```

2. Search through the resultant tsm.sql file for the code defining the SERVERLOG table and within that table, search for all entries containing a hostname entry.

The following SQL content shows an example entry for defining the SERVERLOG table:

```
INSERT INTO SERVERLOG VALUES (6,-1,'2010-10-14 09:50:28','root','\0\0+©é0x
'Ñö?%Çck·',NULL,'NEManager','recreateAssociation',NULL,
'arg1=TL1TSMSNC,arg2=TL1PassThruNoLogin,arg3=null,arg4=telnet://
portnumber=3083/hostname=9.127.97.246/,
arg5=name=TL1TSMSNC,timeout=20,,arg6=null,arg7={OMNI_TL1TAD=tid='\TL1TSMSNC
\','gateway tid='\null\','chatin string='\'.* ACT-USER:TL1TSMSNC:ADMIN:2::PASS;
COMPLD\',' chatout string='\null\','wakeup string='\RTRV-HDR:TL1TSMSNC::1;
COMPLD\',' wakeup attmpts='\1\','wakeup intrvl='\10\','heartbeat string='\RTRV-
HDR:TL1TSMSNC::1;\',' heartbeat response='\COMPLD\','heartbeat intrvl='\300\','
heartbeat timeout='\330\','resync cmd='\RTRV-SNCALM:TL1TSMSNC:ALL:3;\','resync
intrvl='\900\','login list='\null\','logout list='\null\'}',NULL);
```

The part marked in **bold** typeface indicates the entry for connecting to a host.

3. Convert this entry into the following format:

```
<hostname> <portnumber>: "<name>","<chatin_string>"
"<chatout_string>","<wakeup_string>" <wakeup_attempts> <wakeup__intrvl>,
"<heartbeat_string>" "<heartbeat_response>" <heartbeat_intrvl>
<heartbeat_timeout>,"<resync_cmd>" <resync_intrvl>;
```

Note : Ignore the \ ' characters in the database log. If an entry is set to null, change this to "" for the entry in the hosts file.

The following example shows how you would convert the example above into the required format:

```
9.127.97.246 3083: "TL1TSMSNC",".* ACT-USER:TL1TSMSNC:ADMIN:2::PASS; COMPLD"
"", "RTRV-HDR:TL1TSMSNC::1; COMPLD" 1 10,"RTRV-HDR:TL1TSMSNC::1;" "COMPLD"
300 330,"RTRV-SNCALM:TL1TSMSNC:ALL:3;" 900;
```

4. Insert the resultant string into the hosts file specified by the **HostsFile** property.

Data acquisition

Each probe uses a different method to acquire data. Which method the probe uses depends on the target system from which it receives data.

The TL1 Probe acquires data by connecting and logging in to an NE that generates TL1 messages and logging in. It receives the TL1 messages that the NE generates and passes the autonomous messages, responses, and acknowledgements to Netcool/OMNIbus.

The probe sends messages to NEs for heartbeating and resynchronizing connections. This enables the probe to maintain persistent connections with the NEs and to minimize or eliminate lost alarms.

Data acquisition is described in the following topics:

- [“TL1 protocol” on page 4](#)
- [“Issuing commands” on page 4](#)
- [“TL1 command structure” on page 4](#)
- [“Message structure and expected tokens” on page 6](#)
- [“Multiple events per message” on page 6](#)
- [“Peer-to-peer failover functionality” on page 7](#)
- [“Connecting to single or multiple devices” on page 8](#)
- [“Using chat in scripts and chat out scripts” on page 9](#)
- [“Alerting TL1 devices to the probe connection” on page 9](#)
- [“Resynchronizing with the TL1 device” on page 9](#)
- [“Heartbeating” on page 9](#)
- [“Specifying devices using a hosts file” on page 10](#)

- [“Tracking the login status of the network elements on a TL1 device” on page 12](#)
- [“Read timeout” on page 12](#)
- [“Backoff Strategy” on page 12](#)
- [“Data Stream Capture” on page 13](#)
- [“Command line interface” on page 13](#)
- [“Issuing commands to the probe” on page 14](#)

TL1 protocol

TL1 is an ASCII-based management protocol mainly used for managing telecommunication networks.

The protocol specifies the following types of messages:

- Input commands sent to a network element.
- Acknowledgements messages sent from the network element to acknowledge instructions from input command messages.
- Detailed responses from the network element to input commands.
- Autonomous messages sent by network elements either generated periodically, or used to report an unusual event.
- Normal, acknowledge, deny, or partial responses sent by the network elements.

Note : The TL1 Probe processes the TL1 message format as specified in the Bellcore Standard *GR-831-CORE Operations Application Messages- Language for Operations Applications Messages*.

Issuing commands

TL1 commands are issued to the system from a craft terminal (a terminal that performs local maintenance operations) or from an Operations Support System Interface (OSSI).

To issue commands, the user must be logged on to the shelf or node that is the target of the command. The shelf or node that is the target of the command is referred to as the targeted system.

A session is the period of time when a user is logged on to the system. The process of logging on is referred to as *opening a session*. Logging off is referred to as *closing a session*. To open a TL1 session, a user must have a user identifier (UID) and a private identifier (PID, or password). These codes are assigned by the system administrator.

The UID/PID database may be accessed on a per-user basis, so that if a session is inactive for 30 minutes, the system automatically closes the session.

TL1 command structure

Each TL1 command is divided into a set of positional parameters that represent actions or objects. TL1 commands must be typed accurately, with no spaces and with parameters separated by colons.

The following is an example TL1 command line:

```
COMMAND:TID:AID:CTAG:KEYWORD=DOMAIN:STATE;
```

TID, AID, and CTAG are used for routing and controlling the command. The other parameters provide the information to complete the action that the command requests. TL1 commands are not case-sensitive.

Commands, keywords, and domains can be strung together using commas. For example:

```
ED-SYS:TID::CTAG::TMG=EXT1544,AAT=0,ADT=0;
```

[Table 4 on page 5](#) describes some of the TL1 parameters that you can use.

Table 4. TL1 command parameters

Parameter	Description
AID	The AID is an access code used to identify the exact address of the equipment or facilities (channels) within the command.
AID Type	The access identifier (AID) type specifies whether the AID applies to equipment, communications, or one of the traffic or data channels.
CTAG	The correlation tag (CTAG) is a unique identifier that the operator gives to each input command. When the network element responds to that command, it includes the CTAG of the command in the reply. This prevents any confusion about which response applies to which command. CTAGs consist of up to six ASCII characters.
DOMAIN	This parameter contains the list of possible settings for a given keyword. Only one value from the domain can be chosen for each use of the keyword. The proper domain setting must be chosen for the specified keyword when editing the system, equipment, or facilities. Keywords not used in a command are not affected by the command.
PID	The private identifier (PID) is the password associated with the UIS.
SID	The source identifier (SID) is the same as a TID, except that it is used when the network element is identifying itself in a response message.
TID	The target identifier (TID) is a unique name given to each system when it is installed in the network. This name identifies the particular network element to which each command is directed. Each TID comprises a maximum of 20 alphanumeric characters. Note : If the data in the parameter reserved for the TID is not data that has been assigned to any NE as a SID, the command receives a timeout response. When the parameter position reserved for the TID is empty, or null, the parameter defaults to the value placed in that position in the last command entered.
UIS	The user logon name.

Message structure and expected tokens

TL1 messages consist of the following parts: header, identifier, text block, and terminator. Each part has a set of tokens that convey details about the network element and the event being reported.

This section describes the structure of a TL1 message in relation to the expected tokens for each part. The tokens themselves are shown in parentheses in the descriptions that follow.

Header

The header format is the same for both autonomous and output response messages. It contains a terminal identifier (TID), and the date and time that the event occurred.

Identifier

The identifier contains information relevant to the nature of the message. In an autonomous message, this contains an autoid, which identifies the severity of the message. In an output response message, this contains a correlation tag (CTAG), which corresponds to the input command, and a completion code (CompCode).

Text block

The text block is an optional component and contains information that is specific to the message.

Terminator

The terminator indicates the end of the message.

The following example shows a TL1 autonomous message:

```
<cr> <lf> <lf>
^^^BOCALFMA010^03-06-02^01:10:20 <cr> <lf>
*^^456.123^REPT^ALM^T1 <cr> <lf>
^^^"T1AID-16:MN,CGA,NSA" <cr> <lf>
^^^"T1AID-32:<cr> <lf> MN,CGA,NSA" <cr> <lf>
;
```

Where:

- ^^^ are spaces
- BOCALFMA010 is the AID
- 03-06-02 is the date
- 01:10:20 is the time
- 456.123 is the Atag (Autonomous Correlation Tag)
- REPT is the verb
- ALM is an additional modifier to the verb
- T1 is an additional modifier to the verb
- "T1AID-16:MN,CGA,NSA" <cr> <lf> is a quoted line

Multiple events per message

Certain TL1 devices use quoted lines in a text block to send out multiple events in one message. The TL1 probe searches through the text block and pulls out all the quoted lines, comments, and unquoted lines. The probe then sends an alert to the ObjectServer for every quoted line in the text block.

Each alert contains the elements taken from the two header lines and any comments or unquoted lines in the text block. The following example event contains two messages:

```
PTLDORFJDCSNATT001 99-03-14 00:00:01
A 092.284 REPT ALM EQPT::NPC:CGA
/* [0.81001] */
"AR01, MISC ALARM"
/* NPC:<state of alarm>,{CFA|CGA|PBA},MISC,type of alarm */
"649:INFO,CGA,MISC,A"
```

The first message is as follows:

```
$SID -> PTLDORFJDCSNATT001
$Date -> 99-03-14
$Time -> 00:00:01
$AlarmCode -> A
$Atag -> 092.284
$Verb -> REPT
$Modifier1 -> ALM
$Modifier2 -> EQPT::NPC:CGA
$Comment01 -> /* [0.81001] */
$Comment02 -> /* NPC:<state of alarm>,{CFA|CGA|PBA},MISC,type of alarm */
$QuotedLine -> "AR01, MISC ALARM"
```

The second message is as follows:

```
$SID -> PTLDORFJDCSNATT001
$Date -> 99-03-14
$Time -> 00:00:01
$AlarmCode -> A
$Atag -> 092.284
$Verb -> REPT
$Modifier1 -> ALM
$Modifier2 -> EQPT::NPC:CGA
$Comment01 -> /* [0.81001] */
$Comment02 -> /* NPC:<state of alarm>,{CFA|CGA|PBA},MISC,type of alarm */
$QuotedLine -> "649:INFO,CGA,MISC,A"
```

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.

Standard peer-to-peer behavior

By default, the slave probe establishes a connection with, and polls, the TL1 devices and receives the same events as the master probe, but it does not forward the events to the ObjectServer. If the master probe shuts down, the slave probe stops receiving heartbeats from the master probe 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 continues to receive events, but stops sending them to the ObjectServer.

Reducing event traffic

When two TL1 probes are running in a peer-to-peer failover configuration and both are polling devices, there can be an excessive amount of traffic for the client TL1 devices or gateways. This can cause the failure of the TL1 management in the client network.

To reduce event traffic, you can configure the slave probe to not establish a connection with any TL1 device until failover occurs by setting the **SlaveMode** property of the slave probe to 1.

Note : By default, the **SlaveMode** property is set to 0.

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/omnibus.

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    : 5555 # [communication port between master and slave probe]
Mode        : "master"
PidFile     : "$OMNIHOME/var/tl1"

```

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    : 5555 # [communication port between master and slave probe]
Mode        : "slave"
PidFile     : "$OMNIHOME/var/tl1_2"

```

Connecting to single or multiple devices

The TL1 Probe is a multi-headed probe that can connect to either a single device or multiple devices.

When the probe connects to a single device, it uses the details specified by the following properties to make a connection:

- **ChatinString**
- **ChatoutString**
- **HeartbeatInterval**
- **HeartbeatResponse**
- **HeartbeatString**
- **HeartbeatTimeout**
- **Host**
- **Port**
- **ResynchCommand**
- **ResynchInterval**
- **Tid**
- **WakeupAttempts**
- **WakeupInterval**
- **WakeupString**

When the probe connects to multiple devices, it references the hosts file specified by the **HostsFile** property. The hosts file contains the same details as those required for the single connection, but in a different format. For details about this format, see [“Specifying devices using a hosts file” on page 10](#).

If you are using multiple hosts or ports, the probe rotates among the connections, processing one event from each. If the probe experiences a read timeout when doing its first read of any connection in each iteration, it moves directly to the next connection.

The limit to the number of devices to which the probe can connect is determined by the capability of your network and the capability of the host on which you are running the probe. If the processing of your system becomes overloaded when running a probe that monitors multiple TL1 devices, or if there is an appreciable delay between the receipt of a response message and it being sent to the ObjectServer, use more than one probe to monitor your TL1 devices.

Using chat in scripts and chat out scripts

The properties file and command line options use short scripts to control the logging in to and out of the target system.

Chat in and chat out scripts form a single line in the expect-send format (for chat in scripts) or send-expect format (for chat out scripts). The format is:

```
ChatinString : <expect> <send> <expect> <send> ...  
ChatoutString : <send> <expect> <send> <expect>
```

Note : The elements in the chat strings are separated by white space. In order to send or expect a sequence that includes white space, surround the sequence with double quotes.

A typical chat in script might be:

```
".*login.*: .*anu\r\n .*assword.*: .*anu\r\n"
```

Alerting TL1 devices to the probe connection

The TL1 Probe sends a wakeup string property to alert the TL1 device that it has established a connection and is ready to log on.

To specify the wakeup string that the probe sends to TL1 devices, use the **WakeupString** property. Having sent the wakeup string to the device, the probe expects to receive a response message in return. If the probe does not receive the expected regular expression from the TL1, it re-sends the wakeup string. It repeats this for the number of times specified by the **WakeupAttempts** property at intervals specified by the **WakeupInterval** property, or until it receives the expected regular expression. If the probe has still not received the expected response message, it shuts down.

If the probe does receive the expected regular expression from the remote network element, it starts the login process.

Resynchronizing with the TL1 device

After the probe has logged on to a TL1 device, it sends a resynchronization command. This prompts the device to send all its current events to the probe.

To specify what command the probe sends to the TL1 device, use the **ResynchCommand** property. To specify the frequency with which the probe sends resynchronization commands after logging on, use the **ResynchInterval** property.

Heartbeating

When the probe receives no data from the device, heartbeating allows the probe to determine whether there is a problem with either the device or the connection.

If the probe fails to receive any data for the length of time specified by the **HeartbeatInterval** property, the probe sends the string specified by the **HeartbeatString** property to the device. If the probe receives any data, it assumes that both the device and the connection are running correctly. If the probe fails to receive a response within the number of seconds specified by the **HeartbeatTimeout** property, the probe disconnects. When the probe reconnects to the device, it immediately sends a heartbeat string.

If the **HeartbeatInterval** property is set to 0, heartbeating is not used. If the **HeartbeatTimeout** property is set to 0, the probe times out immediately after sending the heartbeat string and goes into reconnection mode.

Specifying devices using a hosts file

The TL1 Probe uses a hosts file when connecting to multiple hosts. This file contains the details that the probe requires to connect to each of your network elements.

This file is specified by the **HostsFile** property. When the probe starts, it reads through the hosts file attempting to connect and log into each host. If the probe encounters a device to which it cannot make a connection to, or log on to, it sends a probe Watch message to the ObjectServer and moves on to the next device in the hosts file.

The hosts file defines the probe connections in the following format:

```
<host> <port> : [<tid>], [ci co], [ws wa wi], [hs hr hi ht], [rc ri];
```

Note : If you encounter a problem uninstalling the probe and you have modified the hosts file, remove the hosts file before uninstalling it again.

Table 5 on page 10 describes the hosts file format and the properties in the properties file that correspond to each item.

Note : The values specified using the **HostsFile** property have a maximum length of 256 characters.

Table 5. Hosts file format		
Item	Corresponding property	Description
host	Host	The host to which the probe connects.
port	Port	The port to which the probe connects.
tid	Tid	The target ID of the TL1 device.
ci	ChatinString	The chat in string for the host/port.
co	ChatoutString	The chat out string for the host/port.
hs	HeartbeatString	The heartbeat string the probe sends to the switch.
hr	HeartbeatResponse	The response the probe expects to the heartbeat request.
hi	HeartbeatInterval	The interval between heartbeats.
ht	HeartbeatTimeout	The time within which the probe expects to receive a response.
rc	ResynchCommand	The command that the probe sends to request resynchronization with the device.
ri	ResynchInterval	The interval with which the probe sends successive resynchronization commands to the device.
ws	WakeupString	The wakeup string that the probe uses to alert the device that it has made a connection and wants to log on.

Table 5. Hosts file format (continued)		
Item	Corresponding property	Description
wa	WakeupAttempts	The maximum number of wakeup attempts that the probe makes before moving on to the next device in the hosts file.
wi	WakeupInterval	The interval with which the probe sends successive wakeup strings to the device.

Be aware of the following syntax rules when creating a hosts file:

- Semicolons (;) indicate the end of each entry in the host file.
- Colons (:) are used to separate the device details from the corresponding connection information.
- Commas (,) separate individual sections within logins.
- Whitespace separates individual elements within the sections.
- Whitespace characters are <sp>, <tab>, <cr>, and <nl>.
- Comment lines are preceded by #.
- Blank lines are ignored.

If the hosts file is defined, all properties in the properties file that define the connection to an individual host are ignored. These are the properties listed in the **Corresponding property** column of [Table 5 on page 10](#).

If the hosts file is not defined, the probe uses the **Host** and **Port** properties to connect to the device. If the **Host**, **Port**, and **HostsFile** properties are not defined, the probe exits.

Format for hosts file when specifying multiple TL1 devices on a single host

The hosts file specifies how the probe connects to the network elements on the target host. The probe is designed to have a dedicated thread for each <hostname> <port> entry in the hosts file, where each thread creates a socket connection towards the host.

The TL1 Probe supports two different formats for the host files.

File format 1

File format 1 uses the following syntax:

```
<host> <port>:
    "<tid1>","ci1" "co1",,,:
    "<tid2>","ci2" "co2",,,:
    "<tid3>","ci3" "co3",,,;
```

Sample from a hosts file using File format 1:

```
host1 10033:
    "NYCM1759","login1 user1\r password pass1\r" "tl1stop\r",,,:
    "NYCM5500","login user2\r password pass2\r" "tl1stop\r",,\r" "HB" 10 15,:
    "NYCM1630",,,;
```

In this example there are three network elements (TIDs) under a single socket connection. After opening a telnet session the probe sends wakeup, login and resync commands which are defined in each TID entry.

File format 2

File format 2 is the recommended format for the hosts file. It contains separate connection entries for each TID element being managed. This maximizes the performance of the probe by making full use of its multi-threaded capabilities.

File format 2 uses the following syntax:

```
<host1> <port1>:  
    "<tid1>","ci1" "co1",,,;  
<host1> <port1>:  
    "<tid2>","ci2" "co2",,,;  
<host1> <port1>:  
    "<tid3>","ci3" "co3",,,;
```

Sample from a hosts file using File format 2:

```
host1 10033:  
    "NYCM1759","login user1\r password pass1\r" "tl1stop\r",,,;  
host1 10033:  
    "NYCM5500","login user2\r password pass2\r" "tl1stop\r",,\r" "HB" 10 15,;  
host1 10033:  
    "NYCM1630",,,;
```

In this example there are three separate socket elements for each of the network elements. After opening the telnet session, the probe only sends wakeup, login and resync commands for a single TID. This maximizes probe performance. The problem, however, is that some element management systems have a limit on the maximum number of connections it can manage at one time. If this is the case with your EMS, use hosts File format 1.

Tracking the login status of the network elements on a TL1 device

For most TL1 devices, there are multiple monitored devices (TIDs) for each connection to a gateway network element. These connections are configured using the hosts file specified by the **HostsFile** property. It is common for a TL1 hosts file to configure up to 30 TIDs on a single connection, each of which requires a login (chat-in) string.

The probe tracks the login status of each TID. When the probe logs in to a TID, it writes a Login Successful ProbeWatch message to its message log. If the login fails to a single TID, whether due to a DENY response or a timeout, the TL1 probe writes a Login Failed ProbeWatch message to its message log and then moves on to the next TID on the TL1 device.

The probe will retry the failed TID connection when it has tried all the other TIDs defined for the TL1 device. It will continue to retry the connection until successful, whereupon it writes a Login Successful ProbeWatch message to its message log.

Read timeout

The **ReadTimeout** property specifies how long the probe waits to read alarm data before timing out. Each time the probe attempts to read an alarm, this is the allotted time that it waits to receive data. If nothing is received, the probe moves on to the next alarm.

Backoff Strategy

Backoff strategy is used when the probe fails to establish a connection or loses an existing connection to the device.

The probe tries to reestablish a connection after one second, two seconds, then four seconds, and so on, up to a maximum of 4096 seconds. Once the connection is made to the specified port, the probe tries to log in to the device. If the probe fails to log in, the probe will move on to log in to the next device in the list (if available) and apply the same backoff strategy to retry the failed log in.

The backoff strategy remains in place until a successful login occurs. The user can also specify a reconnection interval using the **ReconnectionInterval** property or `-reconnectioninterval`

command line option. When this property is enabled, the probe reconnects at the specified time interval instead of using the backoff strategy. If the connection is terminated by the remote host, the connection is also closed by the probe on the host machine.

Data Stream Capture

The TL1 Probe can capture all raw data sent from a device.

The raw data is stored in a log file and can be used for debugging purposes, to develop new features for the probe, or to pass to another application that requires the same data.

To enable stream capture, set the **StreamCaptureFile** property or the `-streamcapturefile` command line option to the full file path of the stream capture file.

The probe creates a new file each time it starts. Each file is differentiated by a timestamp using the following naming convention:

```
streamcapturefile-host:port-time
```

If a reconnection occurs between the TL1 devices and the probe after a disconnection, the probe uses the same stream capture file that was created when the probe started, overwriting all the data kept before the disconnection occurred. You can instruct the probe to append the existing stream capture file by setting the **StreamCaptureFileAppend** property to 1.

Command line interface

The TL1 Probe is supplied with a command line interface (CLI). This interface allows you to perform commands using the probe (for example, to add or delete network elements during runtime). To use the CLI, you must use the **CommandPort** property in the properties file to specify a port through which commands will be sent. When you want to perform commands, use Telnet to connect to this port.

The following table describes the commands that you can use with the command line interface.

Table 6. CLI commands	
Command	Description
bounce	This command closes the management association to the specified NE and then immediately attempts to reopen it.
resynch	This command allows you to resynchronize the given device socket connection to a specific host and port, and requests that a device resends all the data of a particular type that it has sent since it was started.
passthru	This command instructs the probe to send a TL1 message to a TL1 device.
addne	This command adds a network element during runtime.
delne	This command deletes a network element during runtime.
writehost	This command writes details of the current configuration of the network element to a file.

Table 6. CLI commands (continued)	
Command	Description
exit/quit	This command terminates the CLI client session with the probe.

Note : As the CLI is based upon Telnet connections, you can connect to the TL1 Probe from anywhere. This means that simple scripts can be set up to allow users to acknowledge selected events from the Tivoli Netcool/OMNIBus event list by creating desktop tools using Telnet to connect the TL1 Probe, send a command, and then close the connection.

Issuing commands to the probe

You can use the command line interface to issue bounce, resynch, passthru, addne, delne, and writehost commands to the probe.

Bounce command

The bounce command instructs the probe to disconnect from the specified network element and then immediately reconnect.

Note : If you use a `tl1.hosts` configuration file for multiple TIDs under a single `host:port` (that is, using File format 1 as described in [“Format for hosts file when specifying multiple TL1 devices on a single host” on page 11](#)), and if you bounce any one of the TIDs, all of the other TIDs under the same connection will also restart.

The command syntax is:

```
bounce -tid target_id
```

When the command is issued, the probe looks for the connection specified by the **host** and **port** parameters. If found, the probe then closes that connection, which then goes into re-connection mode.

Resynch command

The resynch command instructs the probe to send a resynchronization request to the specified network element. The resynch command that the probe sends is defined in the hosts file.

The command syntax is:

```
resynch -tid target_id
```

When the command is issued, the probe uses the target ID to identify which command in the hosts files to send. For details about the hosts file, see [“Specifying devices using a hosts file” on page 10](#).

Passthru command

The passthru command instructs the probe to send a message to the TL1 device.

The command syntax is:

```
passthru -tid target_id -msg tl1_command
```

Addne command

The addne command adds a network element during runtime.

The command syntax is:

```
addne -host 'host port' -cfg 'configuration'
```

Note : The *configuration* string should use the format described for host file device entries in “Specifying devices using a hosts file” on page 10.

Delne command

The delne command deletes a network element during runtime.

The command syntax is:

```
delne -tid target_id
```

Writehost command

The writehost command writes details of the current configuration of the network element to a log file.

The command syntax is:

```
writehost -file filename
```

Properties and command line options

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

The following table describes the properties and command line options specific to this probe . For information about default properties and command line options, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide* (SC14-7608).

Table 7. Properties and command line options		
Property name	Command line option	Description
ChatinString <i>string</i>	-chatinstring <i>string</i>	Use this property to specify the chat in script for connecting to a target system. The value of this property can be a maximum of 4096 characters long. The default is "". Note : If specifying the ChatinString property using the Hostsfile property the maximum value is 256 characters.
ChatoutString <i>string</i>	-chatoutstring <i>string</i>	Use this property to specify the chat out script for disconnecting from a target system. The value of this property can be a maximum of 4096 characters long. The default is "". Note : If specifying the ChatoutString property using the Hostsfile property the maximum value is 256 characters.

Table 7. Properties and command line options (continued)

Property name	Command line option	Description
CommandPort <i>integer</i>	-commandport <i>integer</i>	Use this property to specify the port number on which the probe listens for commands. For details about using the command line interface, see “Command line interface” on page 13. The default is 0.
ConstantHeartbeat <i>integer</i>	-constantheartbeat <i>integer</i>	Use this property to specify whether the probe uses normal heartbeating mode or constant heartbeating. This property takes the following values: 0: The probe uses normal heartbeating. 1: The probe uses constant heartbeating. The default is 0.
FlushTime <i>integer</i>	-flushtime <i>integer</i>	Use this property to specify the amount of time (in seconds) to wait before flushing a partially read message to the ObjectServer. This is required in case messages arrive that do not have a recognized terminator, in order to prevent the probe from shutting down. The default is 5.
HeartbeatString <i>string</i>	-heartbeatstring <i>string</i>	Use this property to specify the command to send to the network element as a heartbeat. The value that you specify must be a regular expression with a maximum of 4096 characters. The default is "".
HeartbeatInterval <i>integer</i>	-heartbeatinterval <i>integer</i>	When running in normal heartbeating mode, use this property to specify how long (in seconds) the network element is quiet before the probe sends a heartbeat command. When running in constant heartbeating mode, use this property to specify the interval (in seconds) between the sending of successive heartbeats to the network element. The default is 0.

Table 7. Properties and command line options (continued)

Property name	Command line option	Description
HeartbeatResponse <i>string</i>	-heartbeatresponse <i>string</i>	Use this property to specify the response to expect from the device. The value that you specify must be a regular expression with a maximum of 4096 characters. This property is set globally in the properties file, but can be overridden for each network element using the hosts file. The default is "".
HeartbeatTimeout <i>integer</i>	-heartbeattimeout <i>integer</i>	Use this property to specify the time (in seconds) in which to expect a response from the device. The default is 0.
Host <i>string</i>	-host <i>string</i>	Use this property to specify the name of the host to connect to. The value of this property can be a maximum of 4096 characters long. The default is localhost.
HostsFile <i>string</i>	-hostsfile <i>string</i>	Use this property to specify the file that the probe uses to connect to multiple network elements. For details of the format of the hosts file, see “Specifying devices using a hosts file” on page 10. The default is "".
MaxMessageSize <i>integer</i>	-maxmessagesize <i>integer</i>	Use this property to specify the maximum buffer size (in bytes) used by the probe. The recommended value is 4096. The default is 4096.
NoBlankLine <i>integer</i>	-noblankline <i>integer</i>	Use this property to specify whether the probe expects a TL1 message to begin with two newline characters. This property takes the following values: 0: Messages do not begin with two newline characters. 1: Messages begin with two newline characters. The default is 0.

Table 7. Properties and command line options (continued)

Property name	Command line option	Description
PidFile <i>string</i>	-pidfile <i>string</i>	Use this property to specify the name of the PID file that holds the process ID of the probe. The default is \$OMNIHOME/var/tl1.
Port <i>integer</i>	-port <i>integer</i>	Use this property to specify the port number to which the probe connects. The default is 23.
ReadTimeout <i>integer</i>	-readtimeout <i>integer</i>	Use this property to specify the time (in milliseconds) before a socket read times out. The default is 100 (0.1 seconds).
ReconnectionInterval <i>integer</i>	-reconnectioninterval <i>integer</i>	Use this property to specify the time (in seconds) between each connection attempt the probe makes to the network element. For example, if set to 10 it tries to connect every 10 seconds. The default is 0 (the default backoff strategy is implemented).
ResponseTimeout <i>integer</i>	-responsetimeout <i>integer</i>	Use this property to specify the time (in seconds) to wait for a response from the TL1 device when performing chatin and chatout procedures. The default is 20.
ResynchCommand <i>string</i>	-resynchcommand <i>string</i>	Use this property to specify the command to send to the network element in order to resynchronize with the probe. The default is "".
ResynchInterval <i>integer</i>	-resynchinterval <i>integer</i>	Use this property to specify the time interval (in seconds) between the sending of resynchronization commands to the network element. The default is 0.

Table 7. Properties and command line options (continued)

Property name	Command line option	Description
SlaveMode <i>integer</i>	-slavemode <i>integer</i>	Use this property to specify how the slave probe behaves in a peer-to-peer configuration. This property takes the following values: 0: The slave probe establishes connections with the TL1 devices to receive events at the same time as the master probe, but without forwarding any events to the ObjectServer unless the master probe goes down. 1: The slave probe does not establish any connection with the TL1 devices to receive events unless the master probe goes down, thus reducing event traffic. The default is 0.
StreamCaptureFile <i>string</i>	-streamcapturefile <i>string</i>	Use this property to specify the file that is used to store the input stream. If no file is specified then the input stream will not be saved to a file. The default is "".
StreamCaptureFileAppend <i>integer</i>	-streamcapturefileappend <i>integer</i>	Use this property to specify how the probe writes stream capture data following the reconnection of a node. This property takes the following values: 0: Following the reconnection of a node, the probe overwrites the existing StreamCaptureFile. 1: Following the reconnection of a node, the probe appends new stream data to the existing StreamCaptureFile. The default is 0
Tid <i>string</i>	-tid <i>string</i>	Use this property to specify the target identifier of the network element with which to connect. The value of this property can be a maximum of 4096 characters long. The default is "".

Table 7. Properties and command line options (continued)		
Property name	Command line option	Description
UserDefined <i>string</i>	-userdefined <i>string</i>	Use this property to specify a comma-separated list of user-defined properties. These properties will be available to use with the rules file, and are populated whenever an event is generated. The default is "".
WakeupAttempts <i>integer</i>	-wakeupattempts <i>integer</i>	Use this property to specify the number of times to send the wakeup command specified by the WakeupString property. The default is 0.
WakeupInterval <i>integer</i>	-wakeupinterval <i>integer</i>	Use this property to specify the interval (in seconds) between the sending of wakeup tokens. The default is 1.
WakeupString <i>string</i>	-wakeupstring <i>string</i>	Use this property to specify the string sent in order to provoke a response from the device. This string wakes up the device to allow the probe to log in. The value of this property can be a maximum of 4096 characters long. The default is "".

Escape Codes

Some properties support C-style escape codes.

You can use C-style escape codes in the following properties:

- **ChatinString**
- **ChatoutString**
- **HeartbeatString**
- **HeartbeatResponse**
- **WakeupString**

This allows you to easily define whether or not to send escape code sequences after commands. For example, the following ChatinString sends a carriage return character (\r) after the username and password:

```
ChatinString : ".*: user\r .*: passwd\r .*:"
```

The following character sequences are recognized.

Table 8. Escape codes	
Escape code	Character
\b	Backspace

Table 8. Escape codes (continued)	
Escape code	Character
\f	Form-feed
\n	New-line
\r	Carriage return
\t	Tab
\\	Backslash
\'	Single quote
\"	Double quote

Note : The escape sequences for backslash, single quote, and double quote should be double-escaped. For example, to send a backslash character (\) use \\\.

Running the probe

You can run the probe from the command line.

To start the probe , use the following command:

```
$OMNIHOME/tsm/nco_t_t11
```

Note : This is the standard procedure for running any probe . However, if you encounter an error stating that the file cannot be found in \$OMNIHOME/probes/arch, start the probe , using the following command:

```
$OMNIHOME/tsm/nco_t_t11 -propsfile $OMNIHOME/tsm/arch/tl1.props -rulesfile $OMNIHOME/tsm/arch/tl1.rules
```

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 various elements that the probe generates are described in the following topics:

- [“Static elements” on page 21](#)
- [“ProbeWatch Message elements” on page 22](#)
- [“Dynamic elements” on page 22](#)

Static elements

The probe generates the same set of static elements for each event it receives.

The following table describes the elements that the TL1 Probe generates. Not all the elements described are generated for each event; the static elements that the probe generates depends upon the event type.

Table 9. Static elements	
Element name	Element description
\$Host	This element contains the name or IP of the host device to which the probe connects.

<i>Table 9. Static elements (continued)</i>	
Element name	Element description
\$Port	This element contains the port number of the host device to which the probe connects.
\$TID	This element contains the target ID of the switch.
\$Time	This element displays the time that the message was sent.
\$AlarmCode	This element displays the severity level of the message.

ProbeWatch Message elements

The probe includes the \$Host, \$Port, and \$TID elements in all ProbeWatch messages that it generates. The following table describes the elements that the TL1 Probe include in all ProbeWatchmessages that it generates.

<i>Table 10. ProbeWatch message elements</i>	
Element name	Element description
\$Host	This element contains the name or IP of the host device to which the probe connects.
\$Port	This element contains the port number of the host device to which the probe connects.
\$TID	This element contains the target ID of the device.

Dynamic elements

The dynamic elements that the probe generates are entirely dependent on the devices monitored.

The following table describes the dynamic elements that the TL1 Probe can generate. Not all the elements described are generated for each event; the dynamic elements that the probe generates depend upon the event type.

<i>Table 11. Dynamic elements</i>	
Element name	Element description
\$Atag	This element contains the autonomous correlation tag.
\$Modifier1	This element contains an optional modifier to the verb.
\$Modifier2	This element contains an optional modifier to the verb.
\$Verb	This element displays the message type.
\$Ctag	This element displays the response correlation tag.
\$CompCode	This element contains the completion code.

Table 11. Dynamic elements (continued)

Element name	Element description
\$QuotedLine	This element contains the device specific message. There is only one of these.
\$UnquotedLineNN	This element displays the name or position of defined parameters separated by required whitespace.
\$CommentNN	This element displays free format (human readable) text.

Error messages

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

The following table describes the error messages specific to this probe. For information about generic error messages, see the *IBM Tivoli Netcool/OMNIBus Probe and Gateway Guide* (SC14-7608).

Table 12. Error messages

Error	Description	Action
Connect Node not set up	The probe was unable to set up a connect mode.	This is an internal error. Refer to your support contract for details about contacting IBM Software Support.
Connection in unknown state	The connection to the host is in a state that has not been defined in the configuration file.	This is an internal error. Refer to your support contract for details about contacting IBM Software Support.
Date stamp not in recognized format	The date stamp is not in the correct format.	Enter the date stamp in the correct format.
Error initializing list	The probe could not initialize.	This is an internal error. Try running the probe again. If this error recurs, refer to your support contract for details about contacting IBM Software Support.
Expect timed out after number seconds	When the probe tried to log in or log out of the host, the expect part of the chat timed out.	Check that the login name and password are valid for the target host.
Failed to create new command port connection	The probe could not set up a command port.	Check that you have appropriate privileges to run the TL1 Probe.
Failed to initialize TSM	The probe failed to initialize.	This is an internal error. Try running the probe again. If this error recurs, refer to your support contract for details about contacting the IBM Software Support.

Table 12. Error messages (continued)

Error	Description	Action
Failed to set up command port port number	The probe could not start due to a problem setting up or creating the port specified.	Check that the port number specified was positive, that you did not specify a reserved number, and that the port is not in use by another application.
Failed to retrieve value for property CommandPort port number	No command port number was specified in the properties file or on the command line.	Specify a value for CommandPort property, either in the properties file or on the command line.
Failed to set up common TSM properties	The probe could not set up the properties specified in the properties file.	Check that the properties file exists, that the permissions of the file and directory are set correctly, and the properties file is in the correct format.
Failed to start failover - aborting	The failover application failed to start.	Check that both the master and slave are running and that the failover configurations are set correctly on both.
File entry line number: HeartbeatResponse not set!	The hosts file entry specified contains no value for the hr property.	Specify a value for the heartbeat response for the hosts file entry specified.
NMutexLock failed	This is an internal error.	Refer to your support contract for details about contacting IBM Software Support.
Op1RegComp fail for Date	This is an internal error.	Refer to your support contract for details about contacting IBM Software Support.
SID missing from header: string	The source identifier is missing from the message header.	The file received from the host may have become corrupted.
Unknown expect connection error	An unknown error occurred while the probe was receiving the expect part of the chat.	Refer to your support contract for details about contacting IBM Software Support.
Failed to parse auto id Failed to parse header Failed to parse response id Failed to parse text block Failed to parse message Failed to parse quoted line: string	The probe failed to parse the incoming data. The incoming data may be corrupt.	Check that the device is running properly.

Table 12. Error messages (continued)

Error	Description	Action
HeartbeatInterval property may not be negative (value) HeartbeatTimeout property may not be negative (value) ReconnectionInterval may not be negative - reverting to default ResynchInterval property may not be negative (value) WakeupAttempts property may not be negative (value) WakeupInterval property may not be negative (value)	The property specified has been set to a negative value.	Change the property to a positive value.
ReconnectionInterval may not be negative - reverting to default	The property specified has been set to a negative value.	The probe will use the default value for this property.
Unable to send HeartbeatString	The probe could not send a Heartbeat string to the device.	Check that the connection to the host is still active.
Unable to send ResynchCommand	The probe could not send a Resynch command to the device.	Check that the connection to the host is still active.
Unable to compile HeartbeatResponse regexp for: heartbeat response Unable to compile regexp for Heartbeat Response The following host file line did not match the expected format	The device was unable to compile the heartbeat response required by the probe.	Check that the HeartbeatResponse property contains a valid regular expression.
REGEXP: sub number does not exist	A regular expression has not been specified.	Check the rules file and correct the problem.

Table 12. Error messages (continued)

Error	Description	Action
AddNewTextLineValue: Error allocating list space to node AllocateNewConnectNode: Error allocating new node AllocateNewConnectNode: Error allocating new node AllocateNewNode: Error allocating new node LogInOut: Error allocating memory for token SendCommand: Error allocating memory for command WakeUp: Error allocating memory for expect token WakeUp: Error allocating memory for send token	The probe failed to allocate internal storage.	Make more memory available.
Error while logging out of host name port number Error while logging out of host name	The probe failed to log out of the specified device.	Check the connection to the host.
Unexpected syntax for addne command: -host 'string' -cfg 'string'	Invalid syntax was provided with an addne command.	Correct the syntax for the addne command. For details of the correct syntax see “Issuing commands to the probe” on page 14.

ProbeWatch messages

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

The following table describes the raw ProbeWatch error messages that the TL1 Probe generates. For information about generic ProbeWatch messages, see the *IBM Tivoli Netcool/OMNIBus Probe and Gateway Guide* (SC14-7608).

Table 13. ProbeWatch messages

ProbeWatch message	Description	Triggers/causes
Connection Failed ... <i>host_name, port_number,</i> <i>target_id</i>	Failed to connect to one of the hosts specified in the hosts file.	The probe has failed to establish a connection to the host.
Connection Going Down ... <i>host_name,</i> <i>port_number, target_id</i>	The probe is disconnecting from the host.	Either the probe has received a bounce command from the command line, or it is having problems reading from the socket.

Table 13. ProbeWatch messages (continued)

ProbeWatch message	Description	Triggers/causes
Connection Lost ... <i>host_name, port_number, target_id</i>	The probe has lost the connection to the host.	The device has dropped the connection to the probe.
Heartbeat Response Received ... <i>host_name, port_number, target_id</i>	A heartbeat response has been received within the period specified by the HeartbeatTimeout property.	A response matching the HeartbeatResponse property in the properties or hosts file has been received.
Heartbeat Response Timed Out ... <i>host_name, port_number, target_id</i>	A response to the heartbeat string has not been received for the period specified by the HeartbeatTimeout property.	The probe has not received the expected response, or an event, and is disconnecting from and reconnecting to the host.
Heartbeat Sent ... <i>host_name, port_number, target_id</i>	A heartbeat string has been sent from the probe to the host.	The probe has not received an event from the device for the period specified by HeartbeatInterval .
Login Attempted ... <i>host_name, port_number, target_id</i>	The probe is trying to log in to the host.	The probe is trying to log in to the host.
Login Failed ... <i>host_name, port_number, target_id</i>	The probe has failed to log in to the host.	The probe has encountered a problem while logging in. Check that the ChatinString property is specified correctly.
Login Successful ... <i>host_name, port_number, target_id</i>	The probe has logged in to the host.	The probe has logged in to the host.
Wake Up Attempted ... <i>host_name, port_number, target_id</i>	Trying to wake up the network element.	The probe connected to a network element and sent a wakeup command.
Wake Up Failed ... <i>host_name, port_number, target_id</i>	Failed to gain a response from the network element after the specified attempts.	The probe sent a wakeup command to a network element, but failed to received a response.
Wake Up Successful ... <i>host_name, port_number, target_id</i>	The network element has responded to the wakeup command.	A network element responded to a wakeup command sent by the probe.
Going Down	The probe is shutting down.	The probe is shutting down after performing the shutdown routine.
Running	The probe is running normally.	The probe has just been started.

Known issues

This section contains information about known issues.

Specifying the properties and rules files explicitly

When starting the probe , if you encounter an error stating that the file cannot be found in \$OMNIHOME/probes/arch, start the TSM using the following command:

```
$OMNIHOME/tsm/nco_t_tl1 -propsfile $OMNIHOME/tsm/arch/tl1.props -rulesfile  
$OMNIHOME/tsm/arch/tl1.rules
```

Specifying the PID file

You cannot specify the PID file using the **PidFile** property in the properties file. Instead, you must use the -pidfile command line option.

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.



SC14-7631-02

