IBM Tivoli Netcool/OMNIbus Probe for PDS Snyder
6.0

Reference Guide
July 1, 2011

SC23-7911-02

Note

Before using this information and the product it supports, read the information in Appendix A, “Notices and Trademarks,” on page 19.

Edition notice

This edition applies to version 6.0 of IBM Tivoli Netcool/OMNIbus Probe for PDS Snyder (SC23-7911-02) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC23-7911-01.

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 PDS Snyder documentation is provided in softcopy format only. To obtain the most recent version, visit the IBM® Tivoli® Information Center:

http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp?topic=/com.ibm.tivoli.namomnibus.doc/welcome_ptsm.htm

Table 1. Document modification history

Document version	Publication date	Comments
SC23-7911-01	December 31, 2008	Summary table updated. IPv6 support information added. FIPS information added. Installation section added.
SC23-7911-02	July 1, 2011	Information about operating system conventions added in "Conventions used in this guide" on page v. Installation section replaced by "Installing probes" on page 2.

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 PDS Snyder

The Probe for PDS Snyder reads event streams issued by digital cross-connection switches that use the PDS Snyder message protocol, such as the Alcatel DSC CS1L and the Tellabs Titan 532L. It can make connections to multiple devices.

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 PDS Snyder.

Table 3. Summary

Probe target	PDS Snyder
Probe executable name	`nco_t_pds`
Package version	6.0
Probe supported on	For details of supported operating systems, see the following Release Notice on the IBM Software Support website: http://www-01.ibm.com/support/docview.wss?uid=swg21414846
Properties file	`$OMNIHOME/tsm/arch/pds.props`
Rules file	`$OMNIHOME/tsm/arch/pds.rules`
Requirements	A currently supported version of IBM Tivoli Netcool/OMNIbus.
Connection method	TCP/IP
Remote connectivity	The Probe for PDS Snyder can connect to a device on a remote host. When the TSM connects to a single device, it uses chat in and chat out scripts. When the TSM connects to multiple devices, it references a hosts file that contains details of the devices to which connections are required.

Table 3. Summary (continued)	
Multicultural support	Available
Peer-to-peer failover functionality	Available
IP environment	IPv4 and IPv6 **Note :** The probe is supported on IPv6 when running on IBM Tivoli Netcool/OMNIbus 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 *IBM Tivoli Netcool/OMNIbus Installation and Deployment Guide*.

Installing probes

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

The installation process consists of the following steps:

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

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

   http://www-01.ibm.com/support/knowledgecenter/SSSHTQ/omnibus/probes/all_probes/wip/reference/install_download_intro.html
2. Installing the probe using the installation package.

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

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

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

Data acquisition

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

The Probe for PDS Snyder receives a message that reports date, time, and severity. The message also indicates whether it is an alarm, event condition, or a multi-line report. It requires the device to have a serial cable installed from a serial port to the RS-232 port on a terminal server. The TSM then connects to the terminal server through a TCP/IP connection and receives alarms as they are logged to this device.

Reconnection attempts

If the TSM gets disconnected from the system, it tries to reconnect. You can specify how many times the TSM tries to reconnect using the **ReconnectionAttempts** property. If you set this property to 0, the TSM makes unlimited attempts to reconnect to the system.

Bounce and resynchronize

The TSM listens for bounce and resynchronize commands. The bounce command kills the connection to the specified network element and then immediately reconnects. The resynch command re-sends all the alarms from the specified network element since it was started.

Users can run the bounce and resynch commands from the command line from a TCP/IP socket.

Running the bounce command

The **bounce** command allows users to close and re-establish the TSM's connection to a particular device. The command format is:

```
bounce -host hostname -port portnumber
```

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

Running the resynch command

The **resynch** command allows users to force the TSM to send a resynchronization command to a device. The command to be sent is specified in the hosts file entry for that connection. The command format is:

```
resynch -host hostname -port portnumber
```

When the command is issued, the TSM looks for the connection specified by the host and port parameters. If found, the TSM sends the **resynch** command to the specified network element.

Resynchronization

The TSM can make periodic resychronization requests while it is running. To specify the frequency with which it makes these attemps, use the **ResynchInterval** property.

Alternatively, you can specify that the TSM only sends one resynchronization request to the system, at startup. To do this, set the **ResynchInterval** property to 0 and set the **ResynchOnStartUpOnly** property to 1.

Data acquisition is described in the following topics:

Alarm formats

The Probe for PDS Snyder is designed to process messages from the PDS Snyder system. The messages from the switch must be compatible with the PDS Snyder switch format.

Example 1 - DCCS

The following is an example alarm from a PDS DCCS switch:

```
*C15:31:56 01,66 AR01 MISC ALM MJ CGA 284 R COMPL
*C15:33:12 01,69 AR01 MISC ALM MJ CGA 284 Y COMPL
* 15:33:31 01,72 ALM MN ERR DC 284 COMPL
```

Example 2 - Titan 532

The following is an example alarm from a PDS Titan 532 switch:

```
* 17:12:59 10,85 ALM MN ER DC 344 COMPL
*C17:13:07 10,86 AR01 MISC ALM MJ CGA 338 LOF COMPL
I 17:13:07 10,87 AR01 MISC ALM IDLD CGA 338 LOF COMPL
*C17:15:50 10,88 AR01 MISC ALM MJ CGA 344 LOF COMPL
I 17:15:50 10,89 AR01 MISC ALM IDLD CGA 344 LOF COMPL
I 17:15:56 10,90 ALM IDLD ER DC 344 COMPL
* 17:20:06 10,91 ALM MN ER DC 338 COMPL
*C17:21:29 10,92 AR01 MISC ALM MJ CGA 338 LOF COMPL
I 17:21:29 10,93 AR01 MISC ALM IDLD CGA 338 LOF COMPL
I 17:21:35 10,94 ALM IDLD ER DC 338 COMPL
* 17:21:52 10,95 ALM MN ER DC 344 COMPL
*C17:23:52 10,96 AR01 MISC ALM MJ CGA 344 LOF COMPL
I 17:23:52 10,97 AR01 MISC ALM IDLD CGA 344 LOF COMPL
I 17:24:49 10,98 ALM IDLD ER DC 344 COMPL
* 17:27:30 10,99 ALM MN ER DC 338 COMPL
I 17:28:38 10,50 AUD SN CAS 1 CAS 01 04 03 DC 089 FAIL COMPL
I 17:28:40 10,51 AUD SN CAS 1 CAS 01 04 03 DC 093 FAIL COMPL
* 17:30:44 10,52 ALM MN ER DC 344 COMPL
*C17:31:02 10,53 AR01 MISC ALM MJ CGA 338 LOF COMPL
I 17:31:03 10,54 AR01 MISC ALM IDLD CGA 338 LOF COMPL
I 17:31:57 10,55 ALM IDLD ER DC 338 COMPL
*C17:34:14 10,56 AR01 MISC ALM MJ CGA 344 LOF COMPL
I 17:34:15 10,57 AR01 MISC ALM IDLD CGA 344 LOF COMPL
I 17:35:12 10,58 ALM IDLD ER DC 344 COMPL
I 17:36:14 10,59 AUD SN CAS 1 CAS 01 04 03 DC 089 FAIL COMPL
I 17:36:17 10,60 AUD SN CAS 1 CAS 01 04 03 DC 093 FAIL COMPL
* 17:37:52 10,61 ALM MN ER DC 338 COMPL
* 17:41:07 10,62 ALM MN ER DC 344 COMPL
*C17:41:15 10,63 AR01 MISC ALM MJ CGA 338 LOF COMPL
I 17:41:15 10,64 AR01 MISC ALM IDLD CGA 338 LOF COMPL
I 17:42:19 10,65 ALM IDLD ER DC 338 COMPL
I 17:43:29 10,66 AR01 MISC ALM IDLD CGA 344 LOF COMPL
I 17:44:04 10,67 ALM IDLD ER DC 344 COMPL
* 17:48:14 10,68 ALM MN ER DC 338 COMPL
* 17:50:00 10,69 ALM MN ER DC 344 COMPL
```

Chat in and chat out scripts

Chat in and chat out scripts control probe login and logout. These scripts are on a single line in the expect-send format (for chat in scripts) or send-expect format (for chat out scripts). You can specify chat in and chat out strings using the **ChatinString** and **ChatoutString** properties or the `-chatinstring` and `-chatoutstring` command line options.

The format is:

```
ChatinString  : expect send expect send....
```

```
ChatoutString : send expect send expect....
```

Note : Each element in the chat strings is separated by white space. In order to send or expect a sequence that includes white space, surround the sequence with single quotes.

A typical chat in script might be:

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

The expect text can use any regular expression, while the send text can send any characters, including control characters using the standard UNIX/C escape sequences described in “Escape codes” on page 12.

Heartbeating

When the connection is lost between the fault adapter and the device, heartbeating allows the probe to determine whether the device is still alive. If the **Heartbeat** property is set and the target system fails, the probe tries to ping the fault adapter module until it is active again. The probe then reestablishes a connection and starts to acquire alert data again.

If the probe fails to receive any data for any length of time in seconds specified by the **HeartbeatInterval** property, it pings the fault adapter. If the probe does not receive the ping within **HearbeatTimeout** seconds, the probe disconnects and switches to reconnection mode if the **Retry** property is set to `true`.

If the **Retry** property is set to `false`, the probe does not attempt reconnection. The probe disconnects and shuts down.

Backoff strategy

If the **Retry** property is set to `true`, and the probe fails to establish a connection or loses an existing connection to the device, the probe reverts to a backoff strategy. The probe tries to reestablish a connection after one second, two seconds, then four seconds, eight seconds, and so on, up to a maximum of 4096 seconds.

After the connection is made to the specified port, the probe tries to log in to the device. If the probe fails to log in, it shuts down and tries to connect again. 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 remote host terminates the connection, the probe closes the connection on the host machine. The operating system is not allowed to close the connection.

ReadTimeout

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.

Hosts file format

The probe needs a hosts file to connect to multiple hosts. The **HostsFile** property specifies from which file the probe gets host information. The probe reads the hosts file and attempts to connect and log in to each host.

If the **HostsFile** property is defined, the probe attempts to open the specified file. This file must contain the required connection information for the probe in the following format:

```
host port : chi, cho, ft, ia, ri, ra, af;
```

The following table describes the hosts file format.

Table 4. Hosts file format	
Item	**Description**
host	This item specifies the host to which the probe connects.
port	This item specifies the port to which the probe connects.
chi	This item specifies the chat in string for the host/port.
cho	This item specifies the chat out string for the host/port.
ft	This item specifies the flush time value for the host/port.
ia	This item specifies the inactivity alarm for the host/port.
ri	This item specifies the reconnection interval.
ra	This item specifies the reconnection attempts.
af	This item specifies the active flag. You must set this to True. Setting this to False deactivates the connection.

When you create a hosts file, note the following information:

- In the hosts file, the end of each entry is indicated by a semicolon.
- Whitespace and commas separate individual elements within a line.
- Whitespace characters are sp and tab.
- Comment lines are preceded #.
- Blank lines are ignored.
- You can force the probe reread the hosts file without restarting the probe by issuing a USR1 signal. To do this, enter the command kill -USR1 *pid*, where *pid* is the process identifier of the probe.

Data stream capture

The probe can capture the data stream sent from a device in a stream capture file. For each connection, the full data stream is stored in a stream capture file

The data stream is stored using the following naming convention:

streamcapturefile_host_port

where:

- *streamcapturefile* is the value specified by the **StreamCaptureFile** property
- *host* is the name of the host to which the probe is connected
- *port* is the port on which the probe is listening for data

Stream capture data can be used for debugging purposes, to develop new features for the probe, or to pass to other management systems that require the same data.

Note : If you leave the **StreamCaptureFile** property blank, the data stream capture function is disabled.

Rotating stream capture files

The probe can rotate stream capture files; that is, it can write to a stream capture file that is saved and archived periodically when a predefined file size is reached

To use this feature, set the maximum size for the stream capture file using the **MaxStreamCapFileSize** property and set the **DateStreamCapture** property to 1.

When the **DateStreamCapture** property is set to 1, the probe creates a stream capture file with the following naming convention:

streamcapturefile_host_port_date_time

By appending the filename with a timestamp, the probe avoids overwriting the old stream capture file.

The maximum file size specified by the **MaxStreamCapFileSize** property acts as an upper limit for the stream capture file. If the probe reads a stream whose size exceeds the remaining allowable space in the current stream capture file, it saves the current file and creates a new one, thus storing the whole stream in a single file.

If you do not specify a maximum size for the stream capture file, it grows indefinitely until the connection is closed. If the **MaxStreamCapFileSize** property is set and the **DateStreamCapture** property is set to 0, the probe overwrites the stream capture file for that connection each time the maximum file size is reached.

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 TSM. For information about default properties and command line options, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide* (SC14-7608).

Table 5. Properties and command line options

Property name	Command line option	Description
ChatinString *string*	-chatin *string*	Use this property to specify the chat in script for connecting to the target system. The default is " ".
ChatoutString *string*	-chatout *string*	Use this property to specify the chat out script for disconnecting from the target system. The default is " ".
CommandPort *integer*	-commandport *integer*	Use this property to specify the port number to which the TSM listens for bounce and resync commands. The default is 0 (no command port set up for the TSM).
ConstantHeartbeat *integer*	-constantheartbeat *integer*	Use this property to specify whether the TSM runs in Constant Heartbeat mode: 0: The TSM runs in Inactive Heartbeat mode. 1: The TSM runs in Constant Heartbeat mode. The default is 0.

Table 5. Properties and command line options (continued)		
Property name	**Command line option**	**Description**
DateStreamCapture *integer*	`-datestreamcapture` *integer*	Use this property to specify whether the TSM appends a date and time to the stream capture filename: `0`: The TSM does not append the date and time to the stream capture filename. `1`: The TSM appends the date and time to the stream capture filename. The default is `0`. **Note :** Setting the **DateStreamCapture** property to 1 prevents the stream capture file from being overwritten.
HeartbeatInterval *integer*	`-heartbeatinterval` *integer*	Use this property to specify the time (in seconds) that the TSM waits (after receiving data) before sending a HeartbeatString. The default is `0` (the TSM waits indefinitely). **Note :** The length of the **HeartbeatInterval** (which is measured in seconds) must be greater than the length of the **ReadTimeout** (which is measured in milliseconds).
HeartbeatResponse *string*	`-heartbeatresponse` *string*	Use this property to specify the response the TSM expects to receive from the device. This property is a regular expression, with a maximum of 4096 characters. The default is `""`.
HeartbeatString *string*	`-heartbeatstring` *string*	Use this property to specify the command that prompts the switch to give a known response. The default is `""`.
HeartbeatTimeout *integer*	`-heartbeattimeout` *integer*	Use this property to specify the time (in seconds) in which the TSM expects to receive a response from the device. The default is `0` (the TSM does not timeout).
Host *string*	`-host` *string*	Use this property to specify the name of the host to which the TSM connects. The default is `localhost`.

Table 5. Properties and command line options (continued)

Property name	Command line option	Description
HostsFile *string*	`-hostsfile` *string*	Use this property to specify the file that the TSM uses to connect to multiple hosts. The default is `""`.
LoginDelay *integer*	`-logindelay` *integer*	Use this property to specify the delay (in seconds) between the TSM making a connection with a device and beginning the login procedure. The default is `0` (no delay).
MaxStreamCapFileSize *integer*	`-maxstreamcapfilesize` *integer*	Use this property to specify the maximum size (in bytes) of the stream capture file. When this limit is reached, the TSM creates a new file. The default is `0`. **Note :** If the TSM reads a stream whose size exceeds the remaining allowable space in the file, it creates a new one, thus storing the whole stream in a single file.
Port *integer*	`-port` *integer*	Use this property to specify the port to which the TSM connects. The default is 23.
ReadTimeout *integer*	`-readtimeout` *integer*	Use this property to specify the number of milliseconds that the TSM allows the socket to be silent. If this time is exceeded, it assumes that the connection is dead and disconnects. The default is `100`. **Note :** The length of the **ReadTimeout** (which is measured in milliseconds) must be less than the length of the **HeartbeatInterval** (which is measured in seconds).
ReconnectionAttempts *integer*	`-reconnectionattempts` *integer*	Use this property to specify the maximum number of times that the TSM attempts to reconnect to the system. The default is `0` (TSM makes unlimited attempts to reconnect to the system).

Table 5. Properties and command line options (continued)		
Property name	**Command line option**	**Description**
ReconnectionInterval *integer*	`-reconnectioninterval` *integer*	Use this property to specify the interval (in seconds) between successive reconnection attempts. The default is `0` which instructs the TSM to use the standard backoff strategy. For details see "Backoff strategy" on page 5.
ResponseTimeout *integer*	`-responsetimeout` *integer*	Use this property to specify the time (in seconds) the TSM waits for a response from the device when logging in or out. The default is `20`.
ResynchCommand *string*	`-resynchcommand` *string*	Use this property to specify the command the TSM sends to PDS Snyder to request a resynchronization. The default is `""`. **Note :** This command can contain a sequence of commands separated by pause durations.
ResynchInterval *integer*	`-resynchinterval` *integer*	Use this property to specify the time (in seconds) that the TSM waits between successive resynchronization requests. The default is `0` (the TSM does not make any resynchronization requests).
ResynchOnStartUpOnly *integer*	`-noesynchonstartuponly` (equivalent to **ResynchOnStartUpOnly** with a value of `0`) `-resynchonstartuponly` (equivalent to **ResynchOnStartUpOnly** with a value of `1`)	Use this property to specify whether the TSM only makes a single resynchronization request at startup: `0`: The TSM makes periodic resychronization requests. `1`: The TSM makes a single TSM resychronization request at startup. The default is `0`. **Note :** If you set this property to `1`, you must set the **ResynchInterval** property to `0`. The **resynch** command sent on the command port is not affected by this property.
StreamCaptureFile *string*	`-streamcapturefile` *string*	Use this property to specify the file the TSM uses to store the input stream log. Omitting this property disables the stream capture feature. The default is `""`.

Escape codes

You can use C-style escape codes in the **ChatinString** and **ChatoutString** properties. This allows you to easily define whether to send escape code sequences after commands.

For example, the following chatin string sends a carriage return character (\r) after the user name and password:

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

The following table explains the character sequences that are recognized.

Table 6. ChatinString escape codes

Escape code	Character
\b	This escape code specifies the backspace character.
\f	This escape code specifies the form-feed character.
\n	This escape code specifies the new-line character.
\r	This escape code specifies the carriage return character.
\t	This escape code specifies the tab character.
\\	This escape code specifies the backslash character.
\'	This escape code specifies the single quote character.
\"	This escape code specifies the double quote character.

Note : Due to the way in which the above properties are parsed, the escape sequences for backslash, single quote and double quote must be double-escaped. For example, to send a backslash character (\), use \\\\.

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.

Static and dynamic elements are described in the following topics:

- "Static elements" on page 12
- "Dynamic elements" on page 13

Static elements

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

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

Table 7. Static elements	
Element name	**Element description**
$Host	This element displays the network element with which this TSM is communicating.
$Port	This element displays the port on which this TSM is communicating.
$Severity	This element contains the severity indicator. Valid values are: *C: Critical alarm **: Major alarm **P: Major alarm (Tellabs Titan) *: Minor alarm A: Informational message (DSC CS1) I: Informational message (Tellabs Titan)
$Time	This element displays the time the event took place. The format is hh:mm:ss.
$SeqNumber	This element displays the sequence number of the alarm. The format is xx,yy.
$Message	This element displays the event message.

Dynamic elements

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

The following table describes the elements that the Probe for PDS Snyder generates. Not all the elements described are generated for each event; the dynamic elements that the TSM generates depends upon the event type.

Table 8. Dynamic elements	
Element name	**Element description**
$ReportLine01 to *NN*	This element contains the extra lines produced in multi-line reports.
$Timeout	If the connection timed out, this element indicates how long the device was silent before the connection timed out.
$UtlTodDate	This element displays the extra field produced by the UTL TOD DATE COMPL message.

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 TSM. For information about generic error messages, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide* (SC14-7608).

Table 9. Error messages

Error	Description	Action
filename: file not found	The TSM could not find the hosts file specified in the properties file.	Check the **HostsFile** entry in the properties file, and amend if necessary.
Bounce search failed for - *hostname:port_number* Resynch search failed for - *hostname:port_number*	The TSM could not find the connection specified by the host and port parameters of the bounce command.	Check that you have specified the correct hostname and port number.
Error occurred while reading hostfile: *filename* Error reading host file *filename*	The TSM could not open the hosts file for reading.	Check that you have specified the correct filename and that the permissions of the file and directories are set correctly. Amend if necessary.
Expect timed out after *number* seconds	When the TSM tried to log in or out of the host, the expect part of the chat timed out.	Check that the host you are connecting to is running correctly. Check that the login name and password are valid for the target host.
Failed to add line to multi-line end-of-day report - ignoring report	The TSM could not write to the multi-line end of day report and so the report will not be generated.	Check that the permissions on the report allow it to be written to.
Failed to create multi-line end-of-day report - exiting	The TSM failed to generate the multi-line end of day report.	Check that the permissions on the report allow it to be written to.
Failed to create new command port connection	The TSM could not set up a command port.	Check that you have appropriate permissions to run the TSM.
Failed to initialize failover Failed to start failover - aborting	The application failed to start.	Check that both the master and slave are running and that the failover configurations are correctly set on both the master and the slave.
Failed to initialize TSM API	The TSM API failed to initialize. This is an internal error.	Try to rerun the TSM. If this recurs, refer to your support contract for details about contacting IBM Software Support.

Table 9. Error messages (continued)

Error	Description	Action
`Failed to parse message - exiting`	The TSM was unable to parse the message received from the host.	The incoming data may be corrupt. Check that the device is running properly.
`Failed to save PID to file`	The TSM could not save the process ID number to the file specified.	Check the permissions of the PID file and corresponding directories, and amend if necessary.
`Failed to send command` *`command`* `Failed to send` *`message`*	The TSM failed to send the command specified to the device.	Check that the connection to the host is still active.
`Failed to set up command port` *`port`*	The TSM cannot start due to a problem setting up the port specified.	Check that the port number specified was positive, not a reserved number, and that it is not in use by another application.

`HeartbeatInterval property may not be negative (`*`number`*`)`
`HeartbeatTimeout property may not be negative (`*`number`*`)`
`ReconnectionInterval property may not be negative (`*`number`*`)`
`ResynchInterval property may not be negative (`*`number`*`)`
	The property specified has been set to a negative value.	Change the property to a positive value.

`Hostfile entry` *`entry_number`*`: Failed to retrieve host value` `Hostfile entry` *`entry_number`*`: Failed to retrieve port value`	The hosts file entry specified contains no value for the **host** property.	Specify a host name for the hosts file entry specified.
`Hostfile entry` *`entry_number`*`: Chatin property set, but no chatout!` `HostsFile entry` *`entry_number`*`: ResynchCommand property set, but no ResynchInterval!`	The hosts file entry specified contains a value for the **ci** property but no corresponding value for the **co** property.	Specify a chatout string for the hosts file entry specified.
`Hostfile entry` *`entry_number`*`: Parse error ':' expected`	The hosts file entry specified contains no colon.	Add a colon to the hosts file entry specified.

Table 9. Error messages (continued)

Error	Description	Action
`HostsFile entry` *`entry_number`*`: Heartbeat Interval not set!`	The hosts file entry specified contains no value for the **hi** property.	Specify value for the heartbeat interval for the hosts file entry specified.
`HostsFile entry` *`entry_number`*`: Heartbeat Timeout not set!`	The hosts file entry specified contains no value for the **ht** property.	Specify a value for the heartbeat timeout for the hosts file entry specified.
`HostsFile entry` *`entry_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.

`process_chat_string: Error allocating memory for token`
`AllocateNewNode: Error allocating new node`
`ReadHostEntry: Error allocating memory for entry`
`ReadHostEntry: Error allocating memory for line`
`ReadHostFile: Error allocating memory for socket node`
`ReadHostFile: Error allocating memory for token`
	The TSM failed to allocate internal storage.	Make more memory available. If this fails, contact the help desk.

`Read error during chat`	The TSM could not process the response from the host.	Check that the device is running currently.
`Unable to compile HeartbeatResponse regexp for:` *`heartbeat_response`*	The TSM was unable to compile the heart beat response required by the TSM.	Check that the **HeartbeatResponse** property contains a valid regular expression.
`Unable to compile regexp for` *`heartbeat_response`* `Unable to compile regexp for` *`string`*	The device was unable to compile the heart beat response required by the TSM.	Check that the **HeartbeatResponse** property contains a valid regular expression.
`Unable to send HeartbeatString`	The TSM could not send a Heartbeat string to the device.	Check that the connection to the host is still active.
`Unable to send ResynchCommand`	The TSM could not send a **Resynch** command to the device.	Check that the connection to the host is still active.

Table 9. Error messages (continued)		
Error	**Description**	**Action**
`Unknown expect connection error`	An unknown error occurred whilst the TSM was receiving the expect part of a chat script.	Refer to your support contract for details about contacting IBM Software Support.

TSMWatch 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 TSMWatch error messages that the TSM generates. For information about generic TSMWatch messages, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide* (SC14-7608).

Table 10. TSMWatch messages		
Error	**Description**	**Action**
`Connection attempted`	The TSM is trying to establish a connection to the host.	The TSM is trying to establish a connection to the host.
`Connection failed`	The TSM has failed to establish a connection to the host.	General connection failure due either to an incorrect setting of the **`Host`** or **`Port`** property, or the host refusing a connection request from the TSM.
`Connection Going Down …`	The TSM is disconnecting from the host.	Either there are insufficient licenses in the license server, the TSM has received a bounce command from the command port, or the TSM is having problems reading from the socket.
`Connection lost`	The TSM has lost the connection to the host.	The device has dropped the connection to the TSM.
`Connection succeeded`	The TSM has successfully established a connection to the host.	The TSM has successfully established a connection to the host.
`Disconnection Attempted …`	The TSM is attempting to disconnect from the host.	The TSM is shutting down or has lost the connection.
`Disconnection Failed …`	The TSM has failed to disconnect cleanly from the host.	There has been a corruption of the memory location storing the connection details.
`Disconnection succeeded …`	The TSM has disconnected from the host.	The TSM shut down or lost the connection.
`Failed to get a license for host: port`	The TSM is unable to obtain a license for the connection to the host.	Either the license server is down or there are insufficient licenses in the server.
`Going Down`	The TSM is shutting down.	The TSM is shutting down after performing the shutdown routine.

Table 10. TSMWatch messages (continued)

Error	Description	Action
`Heartbeat response received`	A heartbeat response has been received within the period specified by the **`HeartbeatTimeout`** property.	A response matching the **`HeartbeatResponse`** property in the props or hosts file has been received.
`Heartbeat response timed out`	A response to the heartbeat string has not been received for the period specified by the **`HeartbeatTimeout`** property.	The TSM has not received the expected response or an event and is disconnecting from and reconnecting to the host.
`Heartbeat sent`	A heartbeat string has been sent from the TSM to the host.	The TSM has not received an event from the device for the period specified by the **`HeartbeatInterval`** property.
`Login attempted`	The TSM is trying to log in to the host.	The TSM is trying to log in to the host.
`Login failed`	The TSM has failed to log in to the host.	The TSM has encountered a problem while logging in. Check that the **`ChatinString`** property is specified correctly.
`Login succeeded`	The TSM has logged in to the host.	The TSM has logged in to the host.
`Logout attempted`	The TSM is trying to log out from the host.	The TSM has received a command to shut down or has received the bounce command from the command port.
`Logout failed`	The TSM has failed to log out from the host.	The TSM has encountered a problem while logging out. Check that the **`ChatoutString`** property is specified correctly.
`Logout succeeded`	The TSM has logged out from the host.	The TSM has logged out from the host.
`Running …`	The TSM is running normally.	The TSM has just been started.
`Unable to get events`	A problem occurred while receiving events.	Either there was a problem initializing the connection due to insufficient memory or (if this message was sent after some events had been parsed) there was a license or a connection failure.

Appendix A. Notices and Trademarks

This appendix contains the following sections:

- Notices
- Trademarks

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing 2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department 49XA

3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, ibm.com, AIX, Tivoli, zSeries, and Netcool are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Adobe, Acrobat, Portable Document Format (PDF), PostScript, and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

SC23-7911-02