Tivoli Netcool Support's Guide to the Syslog probe

by

Jim Hutchinson

Document release: 2.1

Table of Contents

1Introduction....2

1.1Overview....2
1.2Message Format....3
1.3Date Format....3
1.4Offset Properties....3
1.5QuoteCharacters Property....3
1.6SyslogPIDFile....3
1.7Elements....4
1.8Example Configuration....5
1.8.1Example Message....5

2Rules files....6

2.1Default....6
2.2NcKL....6
2.3Example Property file....6

3Syslog probe environments....7

3.1Linux Syslogd....7
3.1.1Configuring the Probe Server....7
3.1.2Restarting the syslog daemon....8
3.1.3Configuring the Syslog servers....8
3.2Configuring the Syslog probe....9
3.2.1Checking syslog messages....9
3.2.2Peer-to-Peer configuration....10
3.3Language Setting....11
3.4Syslogd probe manual Character reference....12
3.4.1Internationalization support....12
3.4.2Example multi-byte character set on Solaris....12
3.4.3Example multi-byte configuration on Windows....12
3.4.4Supported multibyte character sets....12

1 Introduction

1.1 Overview

The Syslog probe reads a file and creates a set of tokens based upon the expected syslog file syntax as defined by The Internet Engineering Task Force, as documented in RFC 3164 and RFC 5424. There are three ways to read Syslog messages:

nco_p_syslog – log file
nco_p_syslog – FIFO file
nco_p_syslogd – Syslogd replacement

Once configured the probe tokenises the events, line by line, based on the probes properties. Typically problems are seen when the syslog message length is exceeded or if there are issues with the locale character set.

1.2 Message Format

The syslog messages are logged as single lines with an initial timestamp and hostname line header.
Each line is tagged to allow those message types to be extracted.
The maximum number of characters varies but typically is 1024.

```
TimeStamp Host Tag: Message
```

For example:

```
Aug 17 01:02:03 hostname syslogd 1.4.1: restart.
```

1.3 Date Format

The default **TimeFormat property setting is:**

```
date +"%b %d %T"
Aug 21 15:30:38
```

1.4 Offset Properties

OffsetZero defines the start of the event data, the default is 0.
OffsetOne specifes the number of token elements to create, the default is 20.
OffsetTwo specifies the position (count of tokens) within the syslog message at which the details section begins, the default is 6.

```
01234567890123456789012345678901234567890123456789        Characters
1   2  3        4        5       6      7                 Tokens
Aug 17 01:02:03 hostname syslogd 1.4.1: restart.
```

1.5 QuoteCharacters Property

The QuoteCharacters property can be set to adjust the number of tokens, as tokens are defined as strings held between spaces and not inside quote characters.

1.6 SyslogPIDFile

The SyslogPIDFile defines the location of the syslogd processes pid file, which is usually /etc/syslog.pid. This file needs to be readable by the Syslog probe and contains an integer value which is used to send a SIGHUP to the syslogd process on probe start-up.

1.7 Elements

The Syslog probe creates the following tokens:

Element	Description
$Details	This element contains the main body of syslog messages, as specified by properties OffsetZero, OffsetOne, and OffsetTwo.
$Time	This element displays the timestamp of the syslog message, specified as a UNIX time integer.
$Tokennn	This element contains the tokens generated from syslog message, as specified by properties OffsetZero and OffsetOne.
$TokenCount	This element contains the number of tokens created for this event.
$EventCount	This element identifies the sequence number of the events read thus far.
$Raw	The raw syslog message before tokenisation

1.8 Example Configuration

As root create either the fifo file or the log file to read:

```
mknod /var/log/nco-fifo p
```
OR
```
touch /var/log/ncolog
```

Update the syslogd configuration file to write to the chosen log file:
```
vi /etc/rsyslog.conf
# Netcool/OMNIbus logging
*.debug /var/log/ncolog
```

Restart the syslog daemon:
```
/etc/rc.d/init.d/rsyslog restart
```

If the Syslog probe is running during the restart the probe will pick up the restart messages.

1.8.1 Example Message

The 'logger' command can be used to send a test message to the syslog file:

```
logger test message
```

With the probe logging these details:

```
New Kode: Date stamp format [%b %d %T]
Processing   -->
[Event Processor] Raw:       Jul 13 09:27:34 hostname user-id: test message
[Event Processor] Token1:    Jul
[Event Processor] Token2:    13
[Event Processor] Token3:    09:27:34
[Event Processor] Token4:    hostname
[Event Processor] Token5:    user-id:
[Event Processor] Token6:    test
[Event Processor] Token7:    message
[Event Processor] Details:   test message
[Event Processor] TokenCount:        7
[Event Processor] EventCount:        5
[Event Processor] Time:      1468398454
```

where hostname, is the name of the host where the command 'logger' was run, and 'user-id' is the users name.

Other ways to send a message to the syslog file are:

- Invalid user logins
- Using a test file:
  e.g.
  ```
  cat test-syslog-file >> /var/log/ncolog
  ```

2 Rules files

2.1 Default

The default rules file, syslog.rules, uses the following logic:

```
@Class = 200

if (exists($Token4))
{
      @Identifier = "@" + $Token4 + " -> " + $Details
}
@Manager = %Manager
@Summary = $Details
@Node = $Token4
@Severity = 1
@Agent=$Token5
```

The $Token5 is used to identify the syslog message type, which defines the $Details syntax. The rules file uses the Generic Clear fields to provide problem/resolution, and update @Summary to provide meaningful messages for known syslog message types. @FirstOccurrence and @LastOccurrence are set the $Time token.

For ProbeSubSecondId handling it is best to allow the probe to @FirstOccurrence and @LastOccurrence, and set a custom timestamp field to the $Time value.

2.2 NcKL

The Netcool Knowledge Library includes rules file logic for two main vendors:

- Cisco IOS
- Juniper Junos

It also includes ITNM root cause analysis fields set in:

- omnibus36.include.compat.rules
- AdvCorr36.include.compat.rules

2.3 Example Property file

You can set just the Name and RecoveryFile property to start a new probe instance on the same server.
e.g.

```
Server                  : 'NCOMS'
NetworkTimeout          : 15
PollServer              : 30
Name                    : 'syslog_02'
LogFile                 : '/var/log/ncolog'
RulesFile               : '$OMNIHOME/probes/linux2x86/syslog.rules'
RecoveryFile            : '/opt/nrv81/tivoli/netcool/omnibus/var/syslog_02.reco'
#EOF
```

3 Syslog probe environments

The syslog probe can be used locally or remotely, by configuring the UNIX syslog daemon to forward syslog messages to a probe servers syslog daemon or the syslogd probe.

Note : The syslogd probe replaces the syslog daemon and can be used instead of the syslog probe, as it shares the same probe tokenisation features.

3.1 Linux Syslogd

The following is an example of how to configure the default syslog daemon provided with Red Hat linux, although the features will be generally available for other flavours of linux and UNIX. The management of services is varied so the init.d and hard kill methods are given for the syslogd server restart. The TCP port should be used, and it is recommended that a different port is used [not 514] for remote syslog messages.

3.1.1 Configuring the Probe Server

The probe server needs to be configured with a FIFO file and the Syslog TCP listener.

As root create the ncolog fifo file

```
mknod /var/log/nco-fifo p
```

Update the syslogd configuration file to write to the chosen log file and listen on a TCP Port for syslog messages from other syslog daemons.

```
vi /etc/rsyslog.conf
...
# Netcool Syslog:
# Provides TCP Syslog reception
$ModLoad imtcp
$InputTCPServerRun 10514
...
# Netcool/OMNIbus Syslog probe logging
*.debug /var/log/nco-fifo
```

Restart syslogd

Check for LISTENER on the configured port

```
netstat -na | grep 10514
tcp        0      0 0.0.0.0:10514           0.0.0.0:*               LISTEN
tcp6       0      0 :::10514                :::*                    LISTEN
```

You can perform a failed login to check the service is running and logging to syslog.

3.1.2 Restarting the syslog daemon

Use the recommended syslog daemon restart method for your environment.

Root access is required.

System script

```
/etc/rc.d/init.d/rsyslog restart
```

Hard restart
Check for the process id for the syslogd process, and kill the process.

```
ps -ef | grep rsyslogd | grep -v grep
root     1000 ... /usr/sbin/rsyslogd -n

kill -15 1000

ps -ef | grep rsyslogd | grep -v grep

/usr/sbin/rsyslogd -n &

ps -ef | grep rsyslogd | grep -v grep
root     2000 ... /usr/sbin/rsyslogd -n
```

3.1.3 Configuring the Syslog servers

On the other syslog servers the syslog messages need to be forwarded to the probe server host:port [@@ for TCP]

The simple configuration is shown here.

```
File : /etc/rsyslog.conf
# Netcool Syslog:
# Forwarding Syslog messages to a Syslog server as TCP
#
*.*        @@192.168.222.111:10514
```

Where 192.168.222.111 is the probe servers IP Address, and 10514 is the Syslog daemons TCP listening port.

More configurable settings are available, see http://www.rsyslog.com/doc
e.g.

```
# Netcool Syslog:
# Forwarding Syslog messages to a Syslog server with options
#
*.*  action(type="omfwd" target="192.168.222.111" port="10514" protocol="tcp"
           action.resumeRetryCount="100"
           queue.type="linkedList" queue.size="10000")
```

Restart syslogd for the forwarding to begin.

You can use a failed login to check the forwarding is working.

3.2 Configuring the Syslog probe

The FIFO method is recommended for a probe server being used as a syslog message server in a peer-to-peer configuration. The Syslog messages should be forwarded to both the master and slave server, so there will be less need to replay any messages. Equally, if there is a network outage and the syslog messages are not received by either probe server, the latest syslog message will be most useful.

Example property file setting.

```
# Object Server
Server                  : 'AGG_P'
ServerBackup            : 'AGG_B'
# Best practice
NetworkTimeout          : 15
PollServer              : 60
# Buffering
Buffering               : 1
BufferSize              : 200
BufferFlushInterval     : 9
# Use Name to set files for probe instance
Name                    : "syslog_test"
# Set Rules file
RulesFile               : "$NCHOME/omnibus/probes/linux2x86/syslog.rules"
#
# FIFO logging
FifoName                : "/var/log/nco-fifo"
#
# Real file logging
# LogFile               : "/var/log/ncolog"
#
# Other settings
#
# RecoveryFile          : "$OMNIHOME/var/syslog.reco"
# SyslogPIDFile         : "/etc/syslog.pid"
# ReplayFile            : 0
# CleanStart            : 0
# SendHUP               : 0
# TimeFormat            : "%b %d %T"
```

3.2.1 Checking syslog messages

With the FIFO file checking for syslog messages can be done using a number of methods.
Using TCPDUMP will allow the data to be captured.
e.g.
To the command line

```
tcpdump -i any -s 0 port 10514
```

To a file

```
tcpdump -i any -s 0 port 10514 -w $NCHOME/omnibus/var/syslog.pcap
```

You should ensure that the listening port is open in any firewalls and that the probe servers firewall, if enabled, has the port open for TCP too.

Check the FIFO file was created correctly.

```
prw-r--r--  1 root   root         0 Jan 12 12:16 nco-fifo
```

3.2.2 Peer-to-Peer configuration

The Peer-to-Peer settings allow two probes to send events to the object server, regardless of which probe is running, as when both probes are running. The Peerport setting defines a free port on the master probe server where the probe listens, and the PeerHost defines the other probe server on which the probe is running.

Example MASTER probe settings:

```
#
# P2P
#
Mode                    : 'master'
PeerHost                : 'slave_host'
PeerPort                : 9999
BeatInterval            : 10
BeatThreshold           : 5
# Use Name to set files for probe instance
Name                    : 'syslog_master'
# Set Rules file
RulesFile               : '$NCHOME/omnibus/probes/linux2x86/syslog.rules'
```

Example SLAVE Probe settings:

```
#
# P2P
#
Mode                    : 'slave'
PeerHost                : 'master_host'
PeerPort                : 9999
BeatInterval            : 10
BeatThreshold           : 5
#
# Use Name to set files for probe instance
Name                    : 'syslog_slave'
# Set Rules file
RulesFile               : '$NCHOME/omnibus/probes/linux2x86/syslog.rules'
```

3.3 Language Setting

The Syslog probe does not go into much detail on language settings for the Syslog probe, as this is standard.
e.g.
Check the locale settings available using locale.

```
Locale -a
```

Install the required locale if it is missing.

Use the probes environment file to set the probes locale settings.
e.g

```
cd $NCHOME/omnibus/probes/linux2x86
vi nco_p_syslog.env
#! bin/sh
LANG=en_US
LC_ALL=en_US
export LANG LC_ALL
# Check settings
echo "LANG=$LANG"
echo "LC_ALL=LC_ALL"
# EOF
chmod 755 nco_p_syslog.env
```

3.4 Syslogd probe manual Character reference

The Syslogd probe goes into more detail on setting character sets and the supported multibyte character sets. It is provided here for completeness.

3.4.1 Internationalization support

The Syslogd|syslog probe supports multibyte character sets (for example, Japanese) and character sets that contain individual multibyte characters (for example German, French, and Spanish). To view the character sets correctly, you must configure the locale settings on the host machine correctly.
If you are using a language that contains multibyte characters, you must set the LANG environment variables to the name of your character set, and export the LC_ALL environment variable. For example, if you are using Japanese, set these environment variables to ja_JP.UTF-8; if you are using German, set these environment variables to de_DE.UTF-8. This will enable the probe to recognise the multibyte characters used by your character set when they occur in any network events.

3.4.2 Example multi-byte character set on Solaris

The following steps describe how to configure Solaris to use the Japanese character set:
1. Install the necessary components for Japanese on to the host machine using the Solaris CD.
2. Set the LANG and LC_ALL environment variables to ja_JP PCK. This uses SJIS encoding.
Note: You may have to set the LANG in the host machine's default settings file and reboot it to make the changes take effect.
3. Make sure that the file $OMNIHOME/platform/arch/locales/locales.dat has the following entry:

```
locale = ja_JP PCK, japanese, sjis
```

Where ja_JP PCK is the vendor locale, japanese is the Sybase language, and sjis is the Sybase character set.

3.4.3 Example multi-byte configuration on Windows

The following steps describe how to configure Windows to use the Japanese character set:
1. Install the necessary language pack using the Control Panel.
Note: You must reboot the machine to make the character set available.
2. Make sure the file%OMNIHOME%\locales\locales.dat, has the following element:
`locale = jpn, japanese, sjis`
Where jpn is the vendor locale, japanese is the Sybase language, and sjis is the Sybase character set
Note: You must reboot the machine to be able to use the probe as a service in the required locale.

3.4.4 Supported multibyte character sets

Languages	AIX/HP-UX/Linux	Solaris
English(US)	en_US	en_US
Simplified Chinese	zh_CN	zh_CN
Czech	cs_CZ	cs
French (standard)	fr_FR	fr
German (standard)	de_DE	de
Hugarian	hu_HU	hu
Italian (standard)	it_IT	it
Japanese	ja_JP	ja
Korean	ko_KR	ko
Polish	pl_PL	pl
Portuguese(Brazilian)	pt_BR	pt
Russian	ru_RU	ru
Spanish	es_ES	es