

**Tivoli Netcool Support's
Guide to
Nokia NFMP Probe
by
Jim Hutchinson
Document release: 2.1**



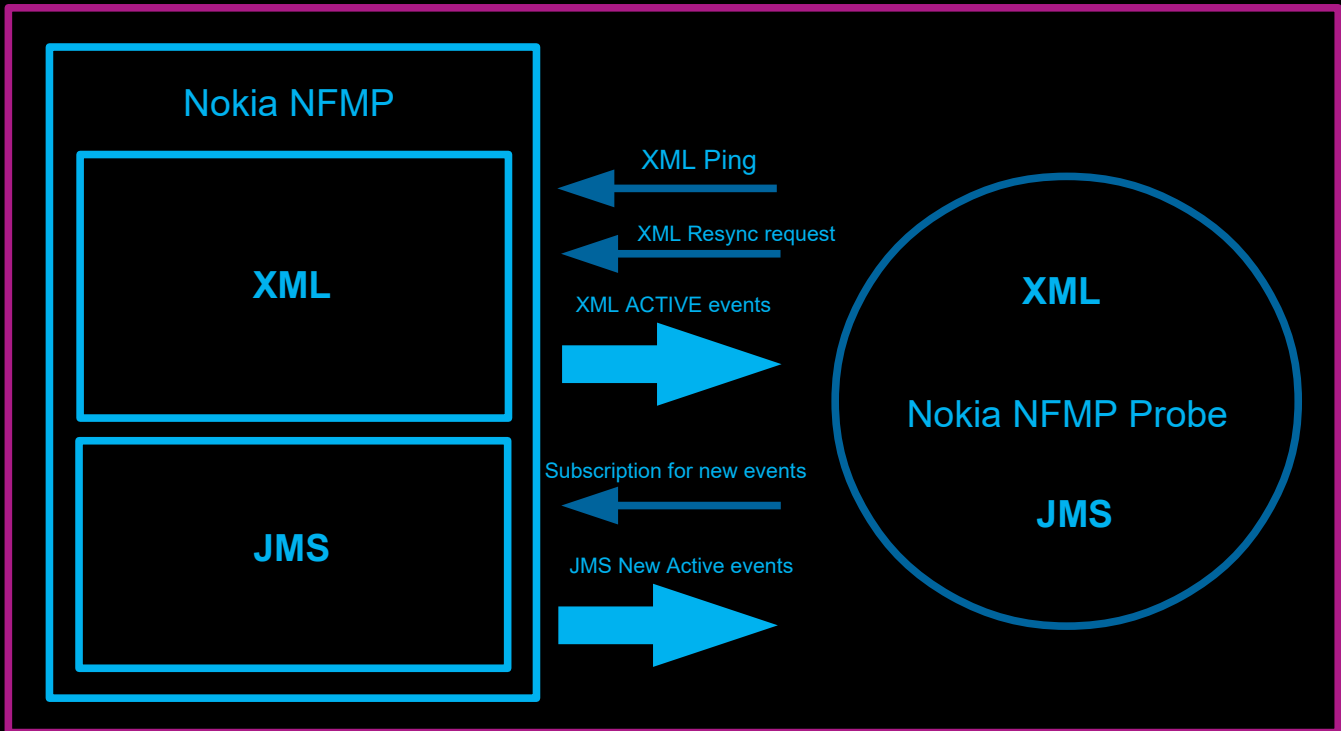
Table of Contents

1Introduction	2
1.1Overview.....	2
2Configuration	3
2.1Nokia NFMP jar file.....	3
2.2Setting NFMPJARHOME and JAVA_HOME.....	3
2.3Process control.....	4
3Probe Property Settings	5
3.1InitialResync property.....	5
3.2ResyncBatchSize property.....	5
3.3The PersistentJmsId property.....	5
3.4Inactivity property.....	5
3.5UseSSL property.....	6
3.6EnableFailover property.....	6
4Nokia NFMP probe Troubleshooting	7
4.1Latest test fix patches.....	7
4.2Nokia NFMP probe resynchronization.....	8
4.3Firewall ports.....	9
4.4Enabling TLS/SSL logging.....	10
5Multiple probes Integration	11
5.1nco_p_nokia_nfmp.wrapper script.....	12

1 Introduction

1.1 Overview

The Nokia NFMP probe replaces the Alcatel 5620 SAM v13 probe, with the probes configuration following close to the original probes settings. The probe synchronises using the HTTP[s]/XML interface and uses the JMS connection for real time active events. Because the probe uses the servers samOss.jar, this file needs to be kept up to date, in the event of any server updates, as does the version of Java being used by the probe.



2 Configuration

The Nokia NFMP probe is configured like the Alcatel 5620 SAM probe.

2.1 Nokia NFMP jar file

The Nokia NFMP probe uses the samOss.jar file from the target Nokia NFMP server to connect to the servers JMS interface. This file needs to be manually copied to the probe before using the probe.

e.g.

```
mkdir $NCHOME/omnibus/probes/java/nco_p_nokia_nfmp
cd $NCHOME/omnibus/probes/java/nco_p_nokia_nfmp
sftp ems-host
username: ems-user
password: *****
ftp> cd /opt/nokia/nfmp/server/nms/integration/SAM_O
ftp> get samOss.jar
ftp> quit
```

2.2 Setting NFMPJARHOME and JAVA_HOME

The NFMPJARHOME is set in the probes environment file as shown:

File : \$NCHOME/omnibus/probes/java/nco_p_nokia_nfmp.env

```
OIDK_PROBE=true

# Set JAVA_HOME to the Java that matches the Nokia NFMP servers java
JAVA_HOME=/opt/java8/jre
export JAVA_HOME

# The samOss.jar file forms a part of the Nokia NSP NFM-P server installation and
# must be copied to the probe server

NFMPJARHOME=${OMNIHOME}/probes/java/nco_p_nokia_nfmp/samOss.jar

# Construct full classpath
CLASSPATH_SETTING=${NFMPJARHOME}

if [ -n "${CLASSPATH}" ]
then
    CLASSPATH=${CLASSPATH_SETTING}:${CLASSPATH}
else
    CLASSPATH=${CLASSPATH_SETTING}
fi

# Extra debug logging
echo "CLASSPATH= $CLASSPATH"
echo "JAVA_HOME= $JAVA_HOME"
echo "NFMPJARHOME= $NFMPJARHOME"
ls -l $NFMPJARHOME
# EOF
```

2.3 Process control

When configuring process control for non-native probes ensure that the `RetryTime` and `RogueTimeout` are set to values equal to or greater than 30 seconds, so that the probes have enough time to start-up and exit. Otherwise the process control backoff strategy will not be engaged and the non-native probes will spin, which will cause most servers problems.

File : `$NCHOME/omnibus/etc/NCO_PA.props`

```
Authenticate      : 'NONE'
Connections       : 100
Name              : 'NCO_PA'
ConfigFile        : '$NCHOME/omnibus/etc/NCO_PA.conf'
#PropsFile        : '$NCHOME/omnibus/etc/NCO_PA.props'
PidFile           : './var/NCO_PA.pid'
# Non-native probe settings
RetryTime         : 30
RogueTimeout      : 60
KillProcessGroup : TRUE
# End of File
```

3 Probe Property Settings

3.1 InitialResync property

The InitialResync property allows the probe to start with just the JMS connection rather than attempting to perform an XML resynchronisation. This is useful when there is a problem with the JMS connection that needs to be debugged. Starting the probe with InitialResync set to 'false' ensures that any log messages relate to the JMS side of event retrieval.

3.2 ResyncBatchSize property

The ResyncBatchSize property can be used to increase the amount of XML data read at time, which is useful, when there are large amounts of events to be read from the Nokia NFMP server during a resynchronisation.

To estimate a valid size, determine the performance of the resynchronisation as a whole, and then divide the amount of data synchronised by a suitable factor.

e.g.

```
100,000 @ default ResyncBatchSize [100] took 300 seconds
100000/100 = 1000 batches
```

From this you would expect a larger ResyncBatchSize setting to improve performance, physical memory permitting.

i.e.

ResyncBatchSize : 500 or 1000

Set the ResyncBatchSize property as required and monitor the probes memory usage, and the time taken to perform the XML data resynchronisation to confirm that the new setting was an improvement.

3.3 The PersistentJmsId property

The PersistentJmsId is the JMS subscription identifier and **must be unique** within the Nokia NFMP ems. The JMS server recommends that the unique identifier is created from the username, client instance and client server IP Address. With the '@' symbol being used as a separator.

e.g.

For a probe host IP Address of 192.168.20.20;

```
PersistentJmsId : "netcool@1@192.168.20.20"
JmsFilter       : "ALA_clientId in ('netcool@1@192.168.20.20','') and ALA_category not in ('STATISTICS','ACCOUNTING')"
```

It is possible to run the probe without a PersistentJmsId;

```
PersistentJmsId : ""
JMSFilter       : "ALA_clientId in ('','')"
```

Note : Always use double-quotes for the properties and single quotes without spaces between commas.

3.4 Inactivity property

The Inactivity property is used to determine if the JMS subscription has fallen silent. The default setting is 0, or indefinite. The object server can be configured to monitor the LastOccurrence of the Nokia NFMP probes events for each system, and report staleness of events as required. Setting the JMS connection to timeout will impact the system, as the probe attempts to recover alarm data through the XML resynchronisation. Therefore, the Inactivity property is typically set to a value of at least an hour [3600 seconds], when it is set.

3.5 UseSSL property

The Nokia NFMP probe can connect to the HTTPS and TLS/SSL JMS port on the Nokia NetAct server. Under these circumstances TLS/SSL trust must be configured between the probe and server. This can usually be achieved through the use of common CA certificate.

```
UseSSL           : "true"
TrustStore       : "/opt/nfmpserver.trustStore"
TrustStorePassword : "newpassword"
CertificateStore  : "/opt/nfmpserver.trustStore"
CertificateStorePassword : "newpassword"
```

Where the probe manual states:

To configure the truststore, use the following steps:

1. Obtain the security certificate from the NFM-P server.
2. Import the security certificate from the NFM-P server.
3. Verify that the security certificate has been imported into the keystore

```
keytool -list -v -keystore /opt/nfmpserver.trustStore -storepass newpassword
Your keystore contains 1 entries
rootca, date, trustedCertEntry,
```

If additional security is required then the Nokia NFMP manual guidelines should be followed, with the probe being considered a Nokia NFMP client.

3.6 EnableFailover property

The Nokia NFMP probe supports failover and failback. The Nokia NFMP servers manage the client connections, with the probe being disconnected from the primary and secondary as required by the servers. The failover/failback can take a significant period, and process control must be configured correctly for the backoff strategy to be used.

```
# Secondary server
EnableFailover       : 'true'
SecondaryHost        : 'host2'
SecondaryEJBPort     : 1099
SecondaryHTTPPort    : 8080
```

For TLS/SSL the store file may require additional certificates, depending upon the security model being used in the Nokia NFMP servers. Refer to the Nokia NFMP manual on how to configure Nokia NFMP clients for TLS/SSL.

4 Nokia NFMP probe Troubleshooting

4.1 Latest test fix patches

PATCH probe-nco-p-nokia-nfmp-2

REVISION 1:

IJ17597 - Prevented probe to exit due to buffer queue full when processing large number of alarms.

PATCH probe-sdk-java-12

REVISION 1

IT30453 - Fixed issue of Generic 3gpp probe reports lost subscription and queue full due to heavy event load during synchronization.

Example version output:

```
Netcool/OMNIBus probe - Version 8.1.0 64-bit
(C) Copyright IBM Corp. 1994, 2012
```

```
Netcool/OMNIBus Probe API Library Version 8.1.0 64-bit
```

Release ID: 2.1.0

```
Jar Build Date: Fri Aug 30 2019 08:51:46 on rhat5es-build1.hursley.ibm.com (Linux
2.6.18-274.17.1.el5)
```

Probe SDK Release ID: 12.1.0

```
Probe SDK Jar Build Date: Wed Oct 09 2019 10:33:01 BST on rhat5es-
build1.hursley.ibm.com (Linux 2.6.18-274.17.1.el5)
```

```
Probe is running on OS: Linux, version: 3.10.0-514.el7.x86_64, arch: x86
```

```
Probe is using IBM Corporation java version: 1.8.0_211, java runtime version:
```

```
8.0.5.35 - pxi3280sr5fp35-20190418_01(SR5 FP35), from this directory :
```

```
/opt/IBM/tivoli/netcool/platform/linux2x86/jre_1.8.0/jre
```

```
API Release ID: 5.50.86
```

```
Library Revisions:
```

```
libnetcool: 5.50.86
```

```
network::ipv6: 5.50.20
```

```
Software Compile Date: Wed Aug 15 07:15:36 UTC 2018 on rhat5es-build1.hursley.ibm.com
(Linux 2.6.18-274.17.1.el5 #1 SMP Wed Jan 4 22:45:44 EST 2012)
```


4.2 Nokia NFMP probe resynchronization

The Nokia NFMP probe sends a request to the Nokia NFMP using 'findFaults' with a filter based on the RecoveryFile's data. This is done to reduce the load generated by the probe on start-up.

To force a resynchronization from EPOCH, which is a full synchronisation, remove the file defined using the probe property RecoveryFile property.

- Stop the probe
- `rm /opt/netcool/omnibus/var/NokiaNFMPRecovery`
- Start the probe.

If the RecoveryFile file does not exist, the probe will send a full resynchronisation request to the Nokia NFMP server.

If events are not being seen in the event list, check the filtering being applied at the probe level by the filter given in the probe property JmsFilter, and the Nokia NFMP servers logs.

Example log file with the recovery file working:

```
RecoveryFile-> /opt/netcool/omnibus/var/NokiaNFMP
Last Recovery data retrieved from /opt/netcool/omnibus/var/NokiaNFMP
Last Time Recovered 1197986201257
Using recovery timestamp 1197986201257
Posting this XML :
<?xml version="1.0" encoding="UTF-8"?><SOAP:Envelope
xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"xmlns:xsi="http://www.w3.org/2001/X
MLSchema-instance"><SOAP:Header><header xmlns="xmlapi_1.0">
<security>
<user>netcool</user>
<password>#####</password>
</security>
<requestID>client1:0</requestID>
</header>
</SOAP:Header>
<SOAP:Body>
<fm.FaultManager.findFaults xmlns="xmlapi_1.0">
<faultFilter>
<and class="fm.AlarmObject"><greater name="lastTimeDetected" value="1197986201257"/>
</and>
</faultFilter>
</fm.FaultManager.findFaults>
</SOAP:Body>
</SOAP:Envelope>
```

The RecoveryFile file is not created automatically and needs to be created manually, after being removed. As the probe user.

```
touch /opt/netcool/omnibus/var/NokiaNFMPRecovery
ls -l /opt/netcool/omnibus/var/NokiaNFMPRecovery
```

4.3 Firewall ports

The Nokia NFMP probe requires three main ports to be opened:

- HTTP port [HTTPPort : **8080**]
- JMS port [EJBPort : **1099**]
- The JMS server port as defined in the Nokia NFMP server (e.g. **8093**)

The systems administrator can confirm which ports are being used and whether or not they are available to the probe server.

4.4 Enabling TLS/SSL logging

The Nokia NFMP probe should be run from the command line when debugging problems.

Enable the TLS/SSL debug logging and extra non-native probe logging to troubleshoot issues with TLS/SSL connectivity.

File : \$NCHOME/omnibus/probes/java/nco_p_nokia_nfmp.env

```
# Debugging
# Create a unique log file name
PROBENAME=`echo $PROGRAM | awk -Fnc_o_p_ '{print $2}'`
UNIQUENAME=${PROBENAME}.$$
echo "UNIQUENAME=${UNIQUENAME}"

# Set debugging variables for SSL
# NCO_JPROBE_JAVA_FLAGS="-Djavax.net.debug=ssl:handshake:verbose $NCO_JPROBE_JAVA_FLAGS"
# FOR ALL
NCO_JPROBE_JAVA_FLAGS="-Djavax.net.debug=all:handshake:verbose $NCO_JPROBE_JAVA_FLAGS"
# Non-native logging
NDE_DEFAULT_LOG_LEVEL="debug"
NDE_FORCE_LOG_MODULE="$NCHOME/omnibus/log/${UNIQUENAME}_forced.log"
NCO_P_NONNATIVE_TRANSCRIPT="$NCHOME/omnibus/log/${UNIQUENAME}_nonnative.log"
export NDE_DEFAULT_LOG_LEVEL NDE_FORCE_LOG_MODULE NCO_P_NONNATIVE_TRANSCRIPT

# EOF
```

You can check the logs for key SSL handshake messages using these grep's.

```
grep '\*\*\*' <ssl.log>
grep -i found <ssl.log>
```

Other checks:

```
grep -i TLS <ssl.log>
grep -i CN= <ssl.log>
```

5 Multiple probes Integration

The `nco_p_nokia_nfmp` wrapper script allows multiple probes to be run by setting the `NFMPJARHOME` variable before running the Nokia NFMP probe.

Each Nokia NFMP probe can use a version specific directory to hold the required jar files:

e.g.

```
$NCHOME/omnibus/probes/java/<JARS_DIRECTORY>
```

Where the `<JARS_DIRECTORY>` is the directory where `samOss.jar` was copied to, from the target Nokia NFMP server.

The `<HOSTSJAR_FILE>` read by the `nco_p_nokia_nfmp` wrapper script contains a list of Host's, as given the probes property file, along with the `<JARS_DIRECTORY>`, separated by a colon [:].

e.g.

```
host1.domain.com:NFMP1
host2.domain.com:NFMP2
host3.domain.com:NFMP3
host4.domain.com:NFMP4
```

For example NFMP1's property file [`NFMP1.props`] uses the `host1.domain.com` hostname.

e.g.

```
Host: "host1.domain.com"
```

The entry for the Nokia NFMP probe in `nco_pa.conf` must include the `-propsfile` definition for the wrapper script to work.

e.g.

```
nco_process 'SAM1'
{
    Command '$OMNIHOME/probes/nco_p_nokia_nfmp.wrapper -propsfile
$OMNIHOME/probes/etc/NFMP1.props' run as 'netcool'
    Host = 'host1.domain.com'
    Managed = True
    RestartMsg = '${NAME} running as ${EUID} has been restored on $
{HOST}.'
    AlertMsg = '${NAME} running as ${EUID} has died on ${HOST}.'
    RetryCount = 0
    ProcessType = PaNOT_PA_AWARE
}
```

5.1 nco_p_nokia_nfmp.wrapper script

```

#!/bin/sh
#####
# This script was written as a wrapper for the
# nco_p_nokia_nfmp probe
#
# It works by using the Host property setting to set the
# NFMPJARHOME as given in a lookup file:
# $NCHOME/omnibus/etc/nokia_nfmp.jars
#
# Whose format is:
# host.domain.com:JARS_DIRECTORY
# Where:
# $NCHOME/omnibus/probes/java/$JARS_DIRECTORY
# is the directory where samOss.jar from the target
# Nokia NFMP is copied to along with any other JARS
#
#####
# Define JARS lookup file
export NFMPJARHOME
export HOSTJARS_FILE VERSIONPROPSFILE GET_PROPS HOST_PROP
#
HOSTJARS_FILE=$NCHOME/omnibus/etc/nokia_nfmp.jars
#
if [ ! -f $HOSTJARS_FILE ]
then
echo " Nokia NFMP HOST:JARDIR file does not exist : $HOSTJARS_FILE"
echo
echo " File format is :-"
echo " HOST1:JARS_DIRNAME1"
echo " HOST2:JARS_DIRNAME2"
echo
echo " * Directory location is : $NCHOME/omnibus/probes/java"
echo " * Minimum JARS are samOss.jar from target Nokia NFMP"
echo " * Ensure RecoveryFile is unique for each probe instance"
echo " * Use the Name property to ensure unique naming"
exit
fi
#
# Get the property files Host details
#
GET_PROPS=0
for arg in ${*}
do
if [ $GET_PROPS -eq 1 ]
then
PROPSFILE=${arg}
if [ -f $PROPSFILE ]
then
HOST_PROP=`grep -v '^#' $PROPSFILE | grep Host | grep -v PeerHost | awk -F: '{ print $2}' | tr -d \" | tr -d \"`
| tail -1`
JARSDIR=`grep $HOST_PROP $HOSTJARS_FILE | awk -F: '{ print $2 }' | tr -d ' '`
fi
fi
case ${arg} in
-propfile)
GET_PROPS=1
;;
esac
done
#
# Must have -propfile defined
#
if [ $GET_PROPS -ne 1 ]
then
echo " -propfile must be defined on the command line"
exit
fi
#
# Set SAMJARHOME
#
SAMJARHOME=$NCHOME/omnibus/probes/java/$JARSDIR
if [ ! -d $SAMJARHOME ]
then
echo " Cannot find SAMJARHOME : $SAMJARHOME"

```

```
exit
else
    if [ -f $SAMJARHOME/samOss.jar ]
    then
        echo " Found samOss.jar"
    else
        echo "*** Did not find samOss.jar"
    fi
fi
echo " Host property is set to $HOST_PROP"
echo " Nokia NFMP JARS Directory was set to [NFMPJARHOME] : "
echo $SAMJARHOME
if [ -f $NCHOME/omnibus/probes/nco_p_nokia_nfmp ]
then
    echo " Attempting to run probe ..."
# Uncomment exec line after testing
exec $NCHOME/omnibus/probes/nco_p_nokia_nfmp $*
else
    echo " File not found : nco_p_nokia_nfmp"
fi
exit
fi
# EOF
```