

**Tivoli Netcool Support's  
Guide to the  
Nokia NetAct v6 Probe  
by  
Jim Hutchinson  
Document release: 2.2**



## Table of Contents

<b>1Introduction</b> .....	<b>2</b>
1.1Overview.....	2
1.2Probe design.....	3
<b>2Probe property file settings</b> .....	<b>4</b>
2.1General best practice settings.....	4
2.2Nokia NetAct connectivity settings.....	4
2.2.1NetAct Server.....	4
2.2.2Entry Point IRP.....	4
2.2.3Naming Service.....	4
2.2.43GPP Objects .....	4
2.3Circular Store and Forward.....	5
2.4ORB Local Settings.....	5
2.5Command port settings.....	5
2.6Character Settings.....	5
2.7Timeout and Inactivity settings.....	6
2.8Unique instances.....	6
2.9NotificationCategories.....	7
2.10Resynch amd Retry.....	7
2.11AlarmFilter and NotificationFilter.....	7
2.12Reducing probe loading.....	7
<b>3Additional Configuration</b> .....	<b>8</b>
3.1Environment File.....	8
3.2 NOTIFY_FM_ALARM_LIST_REBUILT Events.....	8
3.3Command line interface .....	9
<b>4Object Server Performance</b> .....	<b>10</b>
4.1 Object Server tuning.....	10
4.2Aggregation Gateway.....	10
4.3Object Server logging.....	11
<b>5Probe Rules File Logic</b> .....	<b>12</b>
5.1Discarding repeating events.....	12
5.2Host function calls.....	13
5.3Using a discard flag.....	13
5.4Process Automation.....	14
<b>6Latest Fix</b> .....	<b>15</b>
6.1Version output.....	15
6.2APARS.....	15
6.3Non-native Probe .....	16
6.4Java 8.....	16
6.5IBM CORBA Timeouts.....	17
<b>7Generic 3GPP Probe</b> .....	<b>18</b>
7.1Probe property file.....	18
7.2IDLAttrMapFile.....	19
7.3Command line interface.....	20
7.3.1Example Resynchronise script.....	20

# 1 Introduction

## 1.1 Overview

The Nokia NetAct Probe uses the CORBA 3GPP interface to connect to the Nokia NetAct server.

The probe supports the following 3GPP standards:

3GPP Technical Specification	Version	Object
32.111-3	6.2.0	Alarm Integration Reference Point (IRP)
32.303	6.3.0	Notification IRP
32.363	6.3.0	Entry Point IRP

The EMS may support other objects that can interfere with these, so sometimes it is necessary to explicitly set the NotificationCategories property.

The Nokia NetAct server can be accessed using one of three methods:

### 1. The Naming Service

```
NamingServiceHost      :      "<NokiaNetActServer>"
NamingServicePort      :      "<NokiaNetActNamingServerPort>"
```

### 2. The Nokia NetAct Server service:

```
NetActServerName       :      "<NokiaNetActServer>"
NetActServerPort       :      <NokiaNetActServerPort>
```

### 3. The Generic 3GPP Entry Point IOR file:

```
EntryPointIrpFile      :      '<PathToNokiaNetActEPIRPFile>'
```

The Naming Service method is the easiest to administer as the Netcool/OMNIbus dumpns script can be used to check the service is running and the objects available in the service. However, it is less secure and so may not be allowed.

The Entry Point IRP method provides a file with an IOR string that defines the Nokia NetActs Entry Points host and port. This file can be examined using the CORBA script dior, which dumps the IOR string as text.

It is recommended that the latest test fix patch is used as this resolves two issues that are commonly seen with this probe:

Release ID: 5.8.2

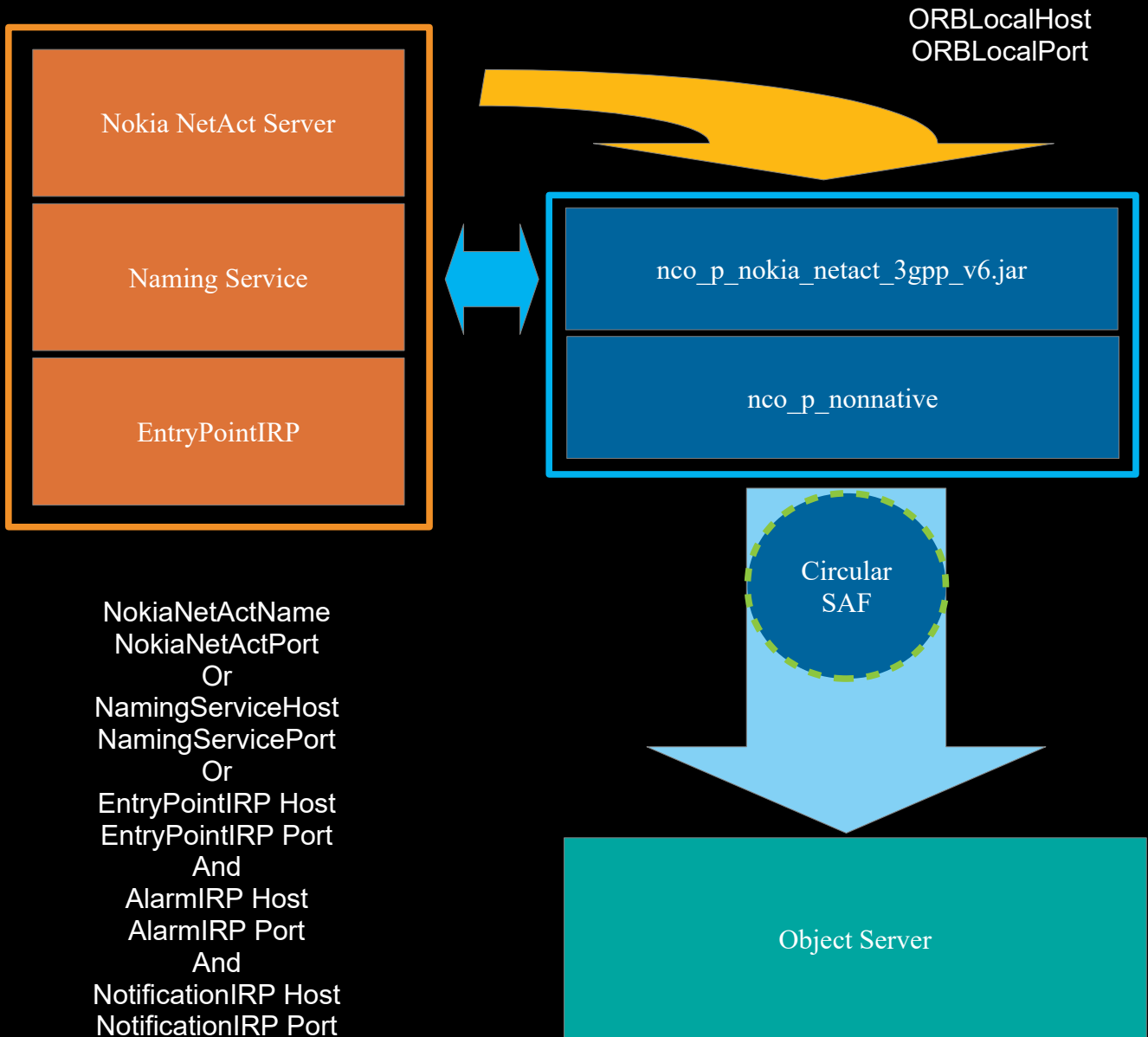
PATCH probe-nco-p-nokia-netact-3gpp-v6-5

REVISION 8

**IV94747** - Fixed issue where probe is unable to complete its subscription attempt on Notification IRP if the first event is NOTIFY\_FM\_ALARM\_LIST\_REBUILT.

**IV94711** - Probe is now allowed to execute resynchronization from command line when Resynch property is set to 'false'.

## 1.2 Probe design



## 2 Probe property file settings

The Nokia NetAct probe supports most of the usual CORBA probe and non-native property file settings. The probe is based on the IBM CORBA API rather than the newer CORBA Framework.

### 2.1 General best practice settings

The following settings are recommended as they reduce the risk of Object Sever connectivity performance issues.

```
# General best practice settings
NetworkTimeout      : 30
PollServer          : 60
# Buffering
Buffering           : 1
BufferSize          : 200
FlushBufferInterval : 9
```

### 2.2 Nokia NetAct connectivity settings

The Nokia NetAct Server can be connected to using one of three methods:

- Nokia NetAct Server
- Entry Point IRP
- Naming Service

The Nokia NetAct Administrator should provide details on which to use.

#### 2.2.1 NetAct Server

```
# NetAct Server:
# NetActServerName      : "<NNHOST>"
# NetActServerPort      : <NNPORT>
```

#### 2.2.2 Entry Point IRP

If the Entry Point IRP IOR file is being used the dior script available in the \$NCHOME/omnibus/probes/java/corba/jacorb-3.3/bin directory can be used to determine the host and port the probe needs to access to connect to the Nokia NetAct Server.

```
# Entry Point
# EntryPointIrpFile      : '$NCHOME/omnibus/var/NetActEPIRP.ior'
```

#### 2.2.3 Naming Service

The Nokia NetAct server can be accessed using the 3GPP naming service.

```
# Naming Service:
NamingServiceHost      : "<NSHOST>"
NamingServicePort      : "<NSPORT>"
```

#### 2.2.4 3GPP Objects

If the Naming service is being used the dumpns script available in the \$NCHOME/omnibus/probes/java/corba directory can be used to determine the object available in the Nokia NetAct Server. Otherwise the object names need to be provided by the Nokia NetAct Administrator.

```
#
# 3GPP Objects from dumpns
#
NotificationIrpName    : 'NOTIFICATIONIRP'
AlarmIrpName           : 'ALARMIRP'
```

## 2.3 Circular Store and Forward

Circular SAF is recommended when there are issues seen with the NSProbe queue limit, which has a fixed limit of 10,000 events. This internal queue can be filled, if there are problems sending events to the object server. When the limit is reached the events are discarded. The root cause of the queue filling up is poor performance at the Object Server, since collection object servers should be capable of receiving the probes event load; Sizing probe servers is discussed in the Netcool/OMNIbus 8.1 manual. Ideally, additional collection layer object servers should be used if the NSProbe queue limit is being exceeded as this behaviour suggests the load on the object server is too large for it to manage.

```
# *****
# Circular Store and Forward settings
# *****
NetworkTimeout      : 5
PollServer          : 60
# Circular SAF settings
StoreAndForward     : 2
SAFFilename         : '$OMNIHOME/var/NokiaNetActProbe'
MaxSAFFileSize      : 10240
SAFPoolSize         : 10
RollSAFInterval     : 90
# Buffering
Buffering           : 1
BufferSize          : 10000
FlushBufferInterval : 3
# *****
# < EoCSAF
# *****
```

## 2.4 ORB Local Settings

The ORB Local settings allow the probe to define the host and port the Nokia NetAct server uses to connect to the probe servers ORB. This is useful in environments where there are firewalls. When these settings are undefined, the probe uses the name given in the naming system for the probe server and a free random port.

```
# ORB Local settings
# ORBLocal property settings
# ORBLocalHostName      : '<PROBESERVER>'
# ORBLocalPort          : '<FREELOCALPORT>'
```

## 2.5 Command port settings

The CommanPort must be set to a free port, and set uniquely for each probe instance on a probe server.

```
# Command port settings
# Unique free port per probe instance
CommandPort          : 5050
CommandPortLimit     : 10
```

## 2.6 Character Settings

The default character settings should be sufficient.

The dumpns script includes the naming services character settings.

```
# Example Character settings
# ORBCharEncoding       : 'ISO8859_1'
# ORBWCharDefault      : 'UTF16'
```

## 2.7 Timeout and Inactivity settings

The TimeTick, Timeout and Agent Heartbeat property settings can be used to control the way in which the probe reacts to event inactivity.

If the TimeTick is set, it defines when the Nokia NetAct Server disconnects the probes subscription.

If the Timeout is set, it defines when the probe disconnects from the Nokia NetAct servers subscription.

The AgentHeartbeat defines the period between checking the probes subscription and can be used alone.

TimeTick is typically 15 minutes by default, in the server, and the Timeout setting must be set to a value greater than the TimeTick value, to allow the probe disconnection to occur after the Nokia NetAct servers disconnection. The AgentHeartbeat needs to be less than both of these settings.

Timeout and AgentHeartbeat are define in seconds, whilst TimeTick is in minutes.

```
# Nokia NetAct Timeout features
# Timeout > TimeTick > Agentheartbeat
#
# TimeTick           : 15
# Timeout            : 960
Agentheartbeat      : 60
```

## 2.8 Unique instances

When there are multiple probes connecting to the Nokia NetAct server, unique Instance and Manager identifiers should be defined to prevent naming allocation conflicts.

```
# Defining a unique Nokia NetAct probe and server instance
#
InstanceId           : "NetActProbe001 <PROBESERVER>"
ManagerIdentifier    : '<NNHOST>_<NNPORT>'
```

## 2.9 NotificationCategories

The Nokia NetAct Probe supports the following 3GPP interfaces:

3GPP Technical Specification	Version	Object
32.111-3	6.2.0	Alarm Integration Reference Point (IRP)
32.303	6.3.0	Notification IRP
32.363	6.3.0	Entry Point IRP

The probes load can be reduced by setting the specific event category's the probe supports in the probe property NotificationCategories.

```
NotificationCategories      : "32.111 V6.2;32.303 V6.3;32.363 V6.3"
```

## 2.10 Resynch and Retry

The affect of probe synchronisation can be managed better by disabling the synchronisation on start-up and performing synchronisation as required using the probes command line interface, typically after the probe has been running for a given period of time and is deemed stable.

Using the Process Controls backoff strategy, of increasing time between retries allows for the Nokia NetAct server to unsubscribe the probes subscription and for subscription timeouts to take affect.

```
Resynch                    : 'false'
Retry                      : 'false'
```

## 2.11 AlarmFilter and NotificationFilter

Both the Alarm and Notification channels support filters whose syntax is defined within the Nokia NetAct server. e.g.

```
AlarmFilter                : "$h==5 or $h==4 or $h==3 or $h==2"
NotificationFilter         : "$type_name=='x1' or $type_name=='x5'"
```

## 2.12 Reducing probe loading

There are not many settings in the probe property file that improve performance outside of the best practice settings for Buffering and NetworkTimeout.

Using MessageLevel 'informational' reduces the probes load due to logging, whilst ensuring there is enough detail in the probes log file to understand the probes behaviour.

Logging RawCapture details increases the load on the operating system, so disabling this should prevent any related issues.

The alerts.details table is usually not used, by the end users, so disabling th creation of detailsdata is recommended. Most probes allow for the setting of the probe property DisableDetails. You can check this for any probe using the -dumpprops option.

```
#
# Performance tunes
#
MessageLevel               : 'info'
RawCapture                 : 0
DisableDetails             : 1
```



## 3 Additional Configuration

The Nokia NetAct probe supports a number of additional configurations.

### 3.1 Environment File

The environment file is used to configure common environment settings, such as the probes locale settings, and debugging options.

```
Directory : $NCHOME/omnibus/probes/java
File : nco_p_nokia_netact_3gpp_v6.env
#! /bin/sh

. $OMNIHOME/probes/java/set_ibm_jre.sh

# IBM CORBA Debug logging
#NCO_JPROBE_JAVA_FLAGS="-Dcom.ibm.CORBA.Debug=true -Dcom.ibm.CORBA.CommTrace=true
-Dcom.ibm.CORBA.Debug.Output=$OMNIHOME/log/nokia_netact_orb_trace.log
$NCO_JPROBE_JAVA_FLAGS"
# Non-native debuglogging
#
#NDE_DEFAULT_LOG_LEVEL="debug"
#NDE_FORCE_LOG_MODULE="$NCHOME/omnibus/log/nokia_netact_nonnative_forced.log"
#NCO_P_NONNATIVE_TRANSCRIPT="$NCHOME/omnibus/log/nokia_netact_nonnative.log"
# Additional CORBA timeout features
#
#NCO_JPROBE_JAVA_FLAGS="-Dcom.ibm.CORBA.AcceptTimeout=1000
-Dcom.ibm.CORBA.AcceptTimeout=5 $NCO_JPROBE_JAVA_FLAGS"
# Locale settings
#
LC_ALL=C
LANG=C
export LC_ALL LANG
# Debug messages
#
echo "LANG=" $LANG
echo "LC_ALL=" $LC_ALL
echo "NCO_JPROBE_JAVA_FLAGS=$NCO_JPROBE_JAVA_FLAGS"
#EOF
```

### 3.2 NOTIFY\_FM\_ALARM\_LIST\_REBUILT Events

A common problem with the CORBA 3GPP probes are the 'NOTIFY\_FM\_ALARM\_LIST\_REBUILT' events, as these cause the probe to resynchronize with the Nokia NetAct server. These events are part of the 3GPP standard, which the probe is configured to. The events can occur for a number of reasons which include service commissioning of nodes, restarting of nodes and Nokia NetAct server requests.

To reduce the impact of unwanted NOTIFY\_FM\_ALARM\_LIST\_REBUILT events the probe can be configured to not perform synchronisation using the probe property "Resync:false". The probe can then be synchronised using the probes command line interface, as required.

For the probe to function correctly with Resynchronization disabled, the latest test fix patch is required.

In addition to this, the probe can be configured, via the probe rules file, to discard events, based on their event time or by their notification or alarm identifier.

### 3.3 Command line interface

The Nokia NetAct probe includes a command line interface which can be connected to using TELNET. The command port needs to be set to a free port on the probe server.

e.g.  
netstat -na | grep 5050  
<nothing returned>

The number of allowed simultaneous connections is defined by the CommandPortLimit.

The probes reads the probe property file on start-up and configures a listener based on the CommandPort value.

e.g.  
CommandPort : 5050  
CommandPortLimit : 10

Once the probe is running the CommandPort shows a TCP/IP listener.

```
netstat -na | grep 5050
... LISTENING
```

The command line interface can then be invoked using the TELNET command.

e.g.  
telnet localhost 5050  
> help  
> resynch\_all

## 4 Object Server Performance

The Object Servers performance affects the way the probe behaves, as without a responsive object server the probe has to wait for the NetworkTimeout to occur before logging issues, and completing tasks.

### 4.1 Object Server tuning

Make sure the following object server tuning is considered.

#### Add Alerts ServerSerial index to all object servers:

```
-- Add indexes for performance
create index AlertsStatusServerSerialIdx on alerts.status (ServerSerial);
```

#### Modify the trigger clean\_details\_table:

```
from
delete from alerts.details where Identifier not in (select Identifier from
alerts.status);
to
delete from alerts.details ;
```

#### Enable the trigger statistics\_cleanup:

Enable trigger

#### Modify the trigger automatic\_backup:

If enable increase the frequency  
Frequency : 30 minutes

### 4.2 Aggregation Gateway

If the probe connects to the aggregation layer ensure that the Aggregation Gateway [AGG\_GATE] is free of any issues. You can run the gateway at message level informational to see if there are any issues and check the 'Resync Manager' log file messages.

Example property settings:

```
MessageLevel           : 'info'
Ipc.Timeout            : 600
Gate.ObjectServerA.Description : 'failover_gate'
Gate.ObjectServerB.Description : 'failover_gate'
Gate.Resync.Enable     : TRUE
Gate.Resync.Type       : 'TWOWAYUPDATE'
Gate.Resync.LockType   : 'PARTIAL'
```

### 4.3 Object Server logging

Review profiler, trigger statistics and self monitoring logs of the Object Servers the probe connects to, and ensure there are no performance issues.

Example checks:

#### Object Server Log:

Object Server error messages:

```
grep Error: AGG_P.log
```

#### Self Monitoring:

Dynamic Table row counts:

```
grep RowCounts: AGG_P_selfmonitoring.log1
```

Alerts messages:

```
grep ALERT: AGG_P_selfmonitoring.log1
```

Number of inserts on dynamic tables:

```
grep InsertCounts: AGG_P_selfmonitoring.log1
```

Profiler logging:

Checking for non-zero timings [can be expanded]:

```
grep Total AGG_P_profiler_report.log1 | grep -v ' 0\.'
```

```
grep Total AGG_P_profiler_report.log1 | grep -v '[0-9]\.'
```

Checking for hidden performance issues [Not 59s or 60s periods]:

```
grep Total AGG_P_profiler_report.log1 | grep -v '(60\.' | grep -v '59\.'
```

Trigger statistics logging:

Checking for non-zero timings [can be expanded]:

```
grep 'for all' AGG_P_trigger_stats.log1 | grep -v ' 0\.'
```

```
grep 'for all' AGG_P_trigger_stats.log1 | grep -v '[0-9]\.'
```

## 5 Probe Rules File Logic

The probe rules file timing is logged using the 'usec' logging message. Typically timings are no more than 100 usec. Where timings exceed 1000 usec, the probes rules file logic should be examined for issue with probe logic and for external calls like gethostaddr and hostname.

As well as using the probe property settings to reduce loading, more fine tuning can be performed within the probe rules file logic using the discard function and suitable probe rules file logic, to prevent discarded events from being processed further once discarded.

### 5.1 Discarding repeating events

It is possible to discard events based on the event time [\$NV\_EVENT\_TIME] recorded in the Nokia NetAct server. In order to identify events you can use the Nokia NetAct servers identifier [\$NV\_ALARM\_ID|\$NV\_NOTIFICATION\_ID] or create an event identifier using the events \$dollar tokens. Then by comparing the stored event time with the new events time, the event can be discarded if found to have already been processed.

```
#####
#DEFINT ARRAYS at the the top of the main rulesfile
#####
array  ALARM_ID_lookup
#####
# New event? get|set ALARMIDTime and process|discard
#####
if ( exists($NV_ALARM_ID) && exists($NV_EVENT_TIME) )
{
    $ALARMIDTime = DateToTime($NV_EVENT_TIME,"%m/%d/%y%n%T%n%p")

#####
# Initialise
#####
    if ( match("",ALARM_ID_lookup[$NV_ALARM_ID]) )
    {
        ALARM_ID_lookup[$NV_ALARM_ID] = $ALARMIDTime
        #log (WARN, "NEW: Stored = " + $NV_ALARM_ID)
    }
#####
# STORED event: Compare times
#####
else
{
    $ALARMIDLookup = ALARM_ID_lookup[$NV_ALARM_ID]
    if ( int($ALARMIDTime) > int($ALARMIDLookup) )
    {
        # log (WARN, "NEW: Store ALARMIDTime = " + $eventTimeLookup + " : " +
$ALARMIDTime )
        @FirstOccurrence = $ALARMIDTime
        @LastOccurrence = $ALARMIDTime
        ALARM_ID_lookup[@Identifier] = $ALARMIDTime
    }
else
{
    log (WARN, "OLD: Discard " + $ALARMIDTime + " <= " + $ALARMIDLookup)
    discard
    $discard_this_event = "true"
}
}
}
}
```

## 5.2 Host function calls

Static variables[%] can be used to store the results from external function calls, like hostname and gethostaddr, to reduce unnecessary loading and improve probe performance.

```
#####
# %STATIC hostlookup at top of main rules file
#####
if(match(%MyHostname,""))
{
    %MyHostname = hostname()
    %MyIPHostname = gethostaddr(%MyHostname)
}
#####
# New command using %STATIC hostlookup
#####
@Manager = @Manager + " probe running on " + %MyHostname

# Old command
#@Manager = @Manager + " probe running on " + hostname()
#
Discarding events based on time
You can discard events based on the Nokia NetAct servers event time to reduce the
volume of events being processed.

#####
# Get ALARMIDTime if not already done
#####
if ( exists($NV_EVENT_TIME) )
{
    $ALARMIDTime = DateToTime($NV_EVENT_TIME,"%m/%d/%y%n%T%n%p")
    $NOW=getdate
#####
# Discard events older than 7 days
#####
if ( int($ALARMIDTime) < (int($NOW) - 604800) )
{
    discard
    $discard_this_event = "true"
}
}
```

## 5.3 Using a discard flag

By setting a discard flag whenever an event is discarded, it is possible to create a rules file where the the main rules file logic can be placed in a separate file, then include this file in the main rules, after checking to see if the event was discarded.

```
#####
# Perform main rules file logic processing
#####
if ( !match($discard_this_event,"true") )
{
    include "main_rules_file_logic.rules"
}
```

## 5.4 Process Automation

The non-native probes need to have an increased RetryTime setting so that they can be restarted using the backoff strategy, where process are started increasingly less frequently:

2, 4, 8, 16, 32, 64, 128, 256 -> 2, 4, 8, 16, 32, 64, 128, 256 ...

Process Automation Property and command line settings:

### KillProcessGroup :

If specified, when the process agent daemon stops a process, it also sends a signal to kill any processes in the same operating system process group. [FALSE]

### RetryTime :

Specifies the number of seconds that a process started by process control must run to be considered a successful start. [5s]

### RogueTimeout :

Specifies the time in seconds to wait for the process to shut down [30s]

For example you can edit the nco script and add additional arguments:

```
vi /etc/init.d/nco
#!/bin/sh
# My Pad settings
MY_ARGUMENTS="-killprocessgroup -roguetimeout 120 -retrytime 60"
...
if [ "$SECURE" = "Y" ]; then
${OMNIHOME}/bin/nco_pad -name ${NCO_PA} -authenticate PAM -secure ${MY_ARGUMENTS} >
/dev/null 2> /dev/null
else
${OMNIHOME}/bin/nco_pad -name ${NCO_PA} -authenticate PAM ${MY_ARGUMENTS} > /dev/null
2> /dev/null
fi
:wq
```

For Netcool/OMNIBus 8.1 you can use the property file:

```
Directory : $NCHOME/omnibus/etc
File : NCO_PA.props
RetryTime: 60
RogueTimeout: 120
KillProcessGroup: TRUE
# End of File
```

## 6 Latest Fix

### 6.1 Version output

```
nco_p_nokia_netact_3gpp_v6 -version
Using IBM Java
Version 1.8.0
Netcool/OMNIBus Nokia Netact 3GPP V6 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: 5.8.2
JAR Build Date: Thu Oct 18 08:53:59 BST 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)
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)
```

### 6.2 APARS

```
PATCH probe-nco-p-nokia-netact-3gpp-v6-5
REVISION 9
IJ29839 - Probe now will reuse its IOR reference on retry to eliminate chances of
creating multiple subscriptions.
- During disconnection, probe will detach all subscription IDs associated with
existing IOR.
IJ30365 - Changed the directory to store probe IOR and subscription ID from /tmp to
$OMNIHOME/var.
REVISION 8
IV94747 - Fixed issue where probe is unable to complete its subscription attempt on
Notification IRP if the first event is NOTIFY_FM_ALARM_LIST_REBUILT.
IV94711 - Probe is now allowed to execute resynchronization from command line when
Resynch property is set to 'false'.
REVISION 7
IV80565 - Added trimming functions on Node field in rules file to avoid leading and
trailing space character.
REVISION 6
IV73385 - If the Timeout property was set, the probe may timeout even while actively
receiving events.
REVISION 5
IV69649 - Added a ManagerIdentifier property which is needed when connecting to the
CORBA interface using EntryPointIrpFile property setting.
REVISION 4
IV66187 - Fixed command port not running after probe reconnection.
REVISION 3
IV62399 - Change ORBCharEncoding property default value from UTF8 to ISO8859_1
REVISION 2
IV61490 - Generate ProbeWatch message after successfully connected to Notification
IRP.
REVISION 1
IV58509 - Allow the probe to attempt reconnection if connection to the notification
service is lost, and the Retry property is set.
```



### 6.3 Non-native Probe

The Non-native probe is an important factor in the performance of the Nokia NetAct probe and the latest 64-bit release should be used.

The latest version is release 21 and this introduced the 64-bit binary:  
e.g.

```
$NCHOME/omnibus/probes/nco_p_nonnative -version  
21.0.0
```

```
find $NCHOME -name nco_p_nonnative  
$NCHOME/omnibus/platform/linux2x86/probes64/nco_p_nonnative  
$NCHOME/omnibus/probes/nco_p_nonnative
```

In the probes -version output you should see these details with the same or higher build date:

```
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)
```

### 6.4 Java 8

The latest version of Java is provided with the Netcool/OMNIbus fix pack.  
e.g.

```
$NCHOME/platform/linux2x86/jre64_1.8.0/jre/bin/java  
$NCHOME/platform/linux2x86/jre_1.8.0/jre/bin/java
```

There is some performance improvement gained when using Java 8 over Java 7.

## 6.5 IBM CORBA Timeouts

There are a few IBM CORBA timeouts that may be useful under extreme circumstances. However, caution needs to be taken to not introduce issues through the use of these property settings. In general the default IBM CORBA settings are suitable, and it is recommended that the cause of the CORBA time out issues are investigated rather than use the IBM CORBA settings.

```
#-Dcom.ibm.CORBA.AcceptTimeout=<s>
# Where <ms> is the wait time in milliseconds. The acceptable range is 0 - 5000.
# If this property is not set, a default value of 0 is used, which prevents a timeout.
# If the property is set but the value that is provided is not valid, a value of 5000 is used
#
#-Dcom.ibm.CORBA.ConnectTimeout=<s>
# Where <s> is the wait time in seconds. The acceptable range is 0 - 300.
# By default, a timeout is not specified.
#
#-Dcom.ibm.CORBA.RequestTimeout=<s>
# Where <s> is seconds in the range 0 - 2147483647. The default value is 0, which prevents timing out.
#
# Short timeouts
#NCO_JPROBE_JAVA_FLAGS="-Dcom.ibm.CORBA.AcceptTimeout=5000 $NCO_JPROBE_JAVA_FLAGS"
#NCO_JPROBE_JAVA_FLAGS="-Dcom.ibm.CORBA.ConnectTimeout=15 $NCO_JPROBE_JAVA_FLAGS"
#NCO_JPROBE_JAVA_FLAGS="-Dcom.ibm.CORBA.RequestTimeout=30 $NCO_JPROBE_JAVA_FLAGS"

# Long timeouts
#NCO_JPROBE_JAVA_FLAGS="-Dcom.ibm.CORBA.AcceptTimeout=5000 $NCO_JPROBE_JAVA_FLAGS"
#NCO_JPROBE_JAVA_FLAGS="-Dcom.ibm.CORBA.ConnectTimeout=300 $NCO_JPROBE_JAVA_FLAGS"
#NCO_JPROBE_JAVA_FLAGS="-Dcom.ibm.CORBA.RequestTimeout=600 $NCO_JPROBE_JAVA_FLAGS"
#
# Needthis to allow control-c/signal to work on timeout
#NCO_JPROBE_JAVA_FLAGS="-Dcom.ibm.CORBA.AllowUserInterrupt=true $NCO_JPROBE_JAVA_FLAGS"
#EOF
```

## 7 Generic 3GPP Probe

The Generic 3GPP probe can be used with the Nokia NetAct server. The benefits of using the Generic 3PP probe over the Nokia NetAct probe are as follows:

Current Probe API : Probe SDK Java

- MaxEventQueueSize – Configurable internal queue size

Current CORBA implementation : CORBA Framework

- jacorb.properties – CORBA properties file

### 7.1 Probe property file

Example property file.

```
Name : 'MyNokiaNetAct'
RulesFile : '$OMNIHOME/probes/linux2x86/MyNokiaNetAct.rules'
NHttpd.AccessLog : '$OMNIHOME/log/MyNokiaNetAct.access.log'
# Object Server
Server : 'AGG_P'
ServerBackup : 'AGG_B'
# Best Practice
NetworkTimeout : 15
PollServer : 60
# Buffering
Buffering : 1
BufferSize : 200
FlushBufferInterval : 9
# Naming Service
NamingServiceHost : '<IP|FQDN>'
NamingServicePort : 15001
Release3GPP : 'V6.3'
IDLAttrMapFile :
'$NCHOME/omnibus/probes/includes/generic_3gpp_v6_3_NokiaNetAct_RuleElementMap.xml'
# Locale
ORBCharEncoding : 'ISO8859_1'
ORBWCharDefault : 'UTF16'
# ORBInitialHost : ''
# ORBInitialPort : ''
#
# Alarms
AlarmIRPName : 'rel_6/AlarmIRP'
NotificationIRPName : 'rel_6/NotificationIRP'
NotificationCategories : '32.111 V6.2;32.303 V6.3;32.363 V6.3'
# NotificationFilter : ''
#
# Inactivity : 0
# Resynch
InitialResynch : 'false'
# ResynchFilter : ''
#
# Command line interface
NHttpd.EnableHTTP : TRUE
NHttpd.ListeningHostname : "localhost"
NHttpd.ListeningPort : 5555
#EOF
```

## 7.2 IDLAttrMapFile

File : \$NCHOME/omnibus/probes/includes/generic\_3gpp\_v6\_3\_NokiaNetAct\_RuleElementMap.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:RuleElement xmlns:tns="http://www.ibm.com/tivoli/netcool/integrations/corbaFramework/RuleElementMap"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ibm.com/tivoli/netcool/integrations/corbaFramework/resource/RuleElementMap
RuleElementMap.xsd" >
  <tns:Idl Category="NotificationIRP" IDLModule="NotificationIRPConstDefs" IDLInterface="AttributeNameValue">
    <tns:Map Element="NotificationID" IdlAttr="NOTIFICATION_ID" />
    <tns:Map Element="EventTime" IdlAttr="EVENT_TIME" />
    <tns:Map Element="SystemDN" IdlAttr="SYSTEM_DN" />
    <tns:Map Element="ManagedObjectClass" IdlAttr="MANAGED_OBJECT_CLASS" />
    <tns:Map Element="ManagedObjectInstance" IdlAttr="MANAGED_OBJECT_INSTANCE" />
  </tns:Idl>
  <!--TS004392620-Connect to Nokia Netact -->
  <tns:Map Element="NV_EVENT_TIME" IdlAttr="EVENT_TIME" />
  <tns:Map Element="NV_MANAGED_OBJECT_CLASS" IdlAttr="MANAGED_OBJECT_CLASS" />
  <tns:Map Element="NV_MANAGED_OBJECT_INSTANCE" IdlAttr="MANAGED_OBJECT_INSTANCE" />
  <tns:Map Element="NV_NOTIFICATION_ID" IdlAttr="NOTIFICATION_ID" />
  <tns:Map Element="NV_SYSTEM_DN" IdlAttr="SYSTEM_DN" />
</tns:Idl>
  <tns:Idl Category="AlarmIRP" IDLModule="AlarmIRPConstDefs" IDLInterface="AttributeNameValue">
    <tns:Map Element="AlarmId" IdlAttr="ALARM_ID" />
    <tns:Map Element="ProbableCause" IdlAttr="PROBABLE_CAUSE" />
    <tns:Map Element="PerceivedSeverity" IdlAttr="PERCEIVED_SEVERITY" />
    <tns:Map Element="SpecificProblem" IdlAttr="SPECIFIC_PROBLEM" />
    <tns:Map Element="AdditionalText" IdlAttr="ADDITIONAL_TEXT" />
    <tns:Map Element="AckTime" IdlAttr="ACK_TIME" />
    <tns:Map Element="AckUserId" IdlAttr="ACK_USER_ID" />
    <tns:Map Element="AckSystemId" IdlAttr="ACK_SYSTEM_ID" />
    <tns:Map Element="AckState" IdlAttr="ACK_STATE" />
    <tns:Map Element="Comments" IdlAttr="COMMENTS"/>
    <tns:Map Element="BackupUpStatus" IdlAttr="BACKED_UP_STATUS" />
    <tns:Map Element="BackupObject" IdlAttr="BACK_UP_OBJECT" />
    <tns:Map Element="ThresholdInfo" IdlAttr="THRESHOLD_INFO" />
    <tns:Map Element="TrendIndication" IdlAttr="TREND_INDICATION" />
    <tns:Map Element="StateChangeDefinition" IdlAttr="STATE_CHANGE_DEFINITION" />
    <tns:Map Element="MonitoredAttributes" IdlAttr="MONITORED_ATTRIBUTES" />
    <tns:Map Element="ProposedRepairActions" IdlAttr="PROPOSED_REPAIR_ACTIONS" />
    <tns:Map Element="CorrelatedNotifications" IdlAttr="CORRELATED_NOTIFICATIONS" />
    <tns:Map Element="Reason" IdlAttr="REASON"/>
    <tns:Map Element="ClearUserId" IdlAttr="CLEAR_USER_ID"/>
    <tns:Map Element="ClearSystemId" IdlAttr="CLEAR_SYSTEM_ID"/>
    <tns:Map Element="AlarmListAlignmentRequirement" IdlAttr="ALARM_LIST_ALIGNMENT_REQUIREMENT"/>
    <tns:Map Element="ServiceUser" IdlAttr="SERVICE_USER"/>
    <tns:Map Element="ServiceProvider" IdlAttr="SERVICE_PROVIDER"/>
    <tns:Map Element="SecurityAlarmDetector" IdlAttr="SECURITY_ALARM_DETECTOR"/>
    <tns:Map Element="VendorSpecificAlarmType" IdlAttr="VENDOR_SPECIFIC_ALARM_TYPE"/>
  </tns:Idl>
  <!--TS004392620-Connect to Nokia Netact -->
  <tns:Map Element="CLEAR_SYSTEM_ID" IdlAttr="CLEAR_SYSTEM_ID"/>
  <tns:Map Element="CLEAR_USER_ID" IdlAttr="CLEAR_USER_ID"/>
  <tns:Map Element="NV_ACK_STATE" IdlAttr="ACK_STATE" />
  <tns:Map Element="NV_ACK_SYSTEM_ID" IdlAttr="ACK_SYSTEM_ID" />
  <tns:Map Element="NV_ACK_TIME" IdlAttr="ACK_TIME" />
  <tns:Map Element="NV_ACK_USER_ID" IdlAttr="ACK_USER_ID" />
  <tns:Map Element="NV_ADDITIONAL_TEXT" IdlAttr="ADDITIONAL_TEXT" />
  <tns:Map Element="NV_ALARM_ID" IdlAttr="ALARM_ID" />
  <tns:Map Element="NV_ALARM_LIST_ALIGNMENT_REQUIREMENT" IdlAttr="ALARM_LIST_ALIGNMENT_REQUIREMENT"/>
  <tns:Map Element="NV_BACK_UP_OBJECT" IdlAttr="BACK_UP_OBJECT" />
  <tns:Map Element="NV_BACKED_UP_STATUS" IdlAttr="BACKED_UP_STATUS" />
  <tns:Map Element="NV_COMMENTS" IdlAttr="COMMENTS"/>
  <tns:Map Element="NV_CORRELATED_NOTIFICATIONS" IdlAttr="CORRELATED_NOTIFICATIONS"/>
  <tns:Map Element="NV_MONITORED_ATTRIBUTE" IdlAttr="MONITORED_ATTRIBUTES"/>
  <tns:Map Element="NV_PERCEIVED_SEVERITY" IdlAttr="PERCEIVED_SEVERITY" />
  <tns:Map Element="NV_PROBABLE_CAUSE" IdlAttr="PROBABLE_CAUSE" />
  <tns:Map Element="NV_PROPOSED_REPAIR_ACTIONS" IdlAttr="PROPOSED_REPAIR_ACTIONS" />
  <tns:Map Element="NV_REASON" IdlAttr="REASON"/>
  <tns:Map Element="NV_SECURITY_ALARM_DETECTOR" IdlAttr="SECURITY_ALARM_DETECTOR"/>
  <tns:Map Element="NV_SERVICE_PROVIDER" IdlAttr="SERVICE_PROVIDER"/>
  <tns:Map Element="NV_SERVICE_USER" IdlAttr="SERVICE_USER"/>
  <tns:Map Element="NV_SPECIFIC_PROBLEM" IdlAttr="SPECIFIC_PROBLEM" />
  <tns:Map Element="NV_STATE_CHANGE_DEFINITION" IdlAttr="STATE_CHANGE_DEFINITION" />
  <tns:Map Element="NV_THRESHOLD_INFO" IdlAttr="THRESHOLD_INFO" />
  <tns:Map Element="NV_TREND_INDICATION" IdlAttr="TREND_INDICATION" />
  <tns:Map Element="NV_VENDOR_SPECIFIC_ALARM_TYPE" IdlAttr="VENDOR_SPECIFIC_ALARM_TYPE"/>
</tns:Idl>
</tns:RuleElement>
```

## 7.3 Command line interface

The new probe API uses the `nco_http` command to run commands in the probe. This interface can be secured using HTTPS and a username/password. The simplest way to secure the port is to listen using localhost and only allow commands to be run locally, which includes by a local `nco_pad` process, if required.

```
# Command line interface
NHttpd.EnableHTTP           : TRUE
NHttpd.ListeningHostname    : "localhost"
NHttpd.ListeningPort        : 5555
```

### 7.3.1 Example Resynchronise script

```
nco_http_resynch_all.sh
#!/bin/sh
# Check arguments
if [ $# -ne 2 ]
then
    echo "Usage : `basename $0` [host] [port]"
    exit
fi
# Set arguments
HOST=$1
PORT=$2
export HOST PORT

# Resynchronise all
$NCHOME/omnibus/bin/nco_http -uri http://${HOST}:${PORT}/probes/generic_3gpp -datatype application/JSON -method
POST -data '{"command":"resynch_all","params":[]}'

# Resynchronise with filter
$NCHOME/omnibus/bin/nco_http -uri http://${HOST}:${PORT}/probes/generic_3gpp -datatype application/JSON -method
POST -data '{"command":"resynch_filter", "params":[{"resync_filter":"$h==1"}]}'
#EOF
```