Tivoli Netcool Support's Guide to the Huawei U2000 CORBA probe

by

Jim Huchinson

Document release: 2.0

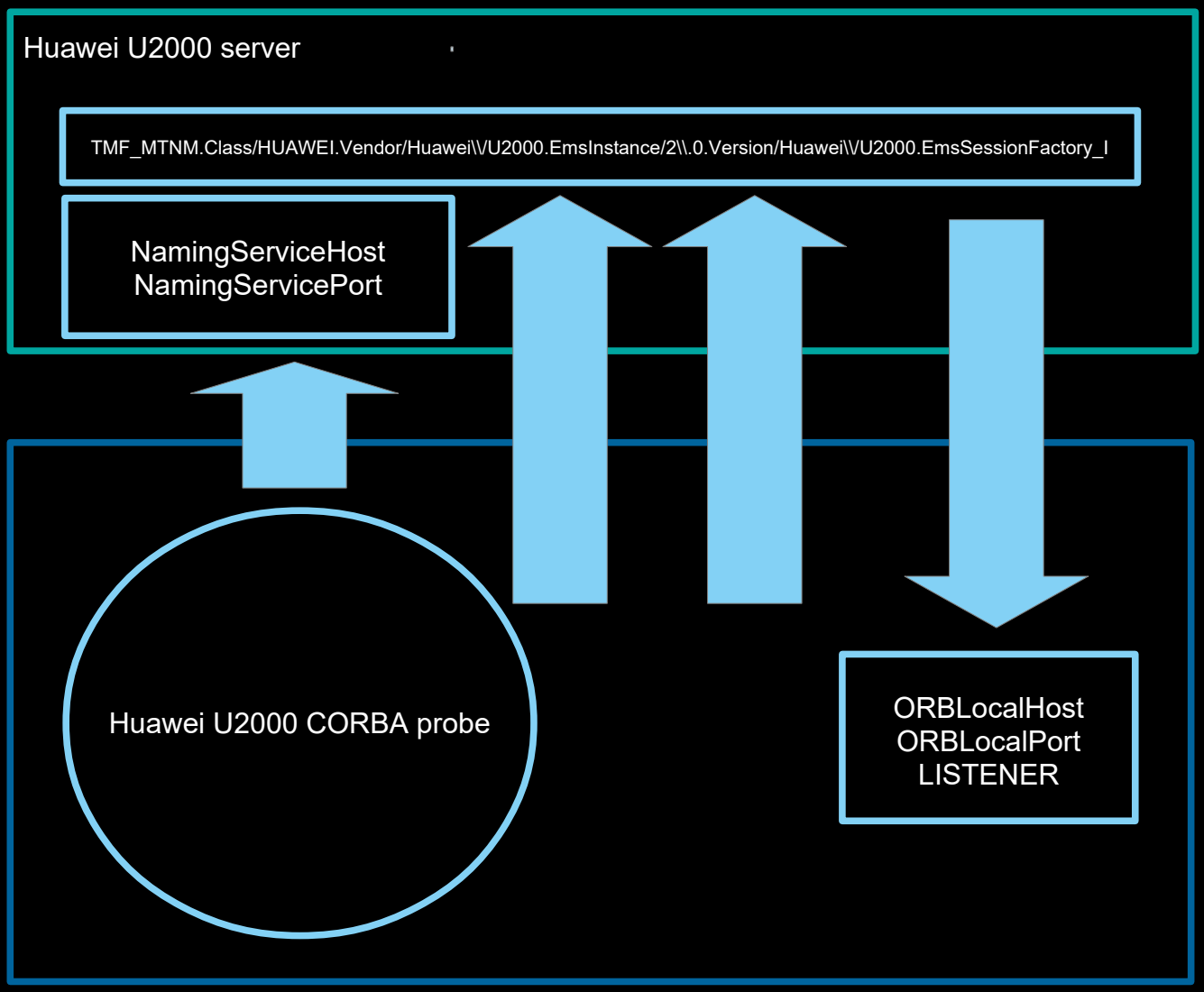
Table of Contents

1 Introduction

1.1 Overview

The Huawei U2000 CORBA probe connects to the Huawei U2000 EMS using the TMF814 CORBA interface running on the Huawei U2000 iManager server.

The Probes manual covers the main configuration of the Huawei U2000 CORBA probe and should be referred to. This document is provided as a supplement to the product documentation.



1.2 Latest version

There are test fix patches available for the probe and supporting packages to resolve load issues on start-up.

omnibus-probe-nco-p-huawei-u2000-corba-1.4.3.0.zip
omnibus-probe-sdk-java-1.12.1.0.zip
omnibus-probe-corba-framework-1.10.1.0.zip

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: 4.3.0
Jar Build Date: Wed Feb 26 2020 06:44:48 on rhat5es-build1.hursley.ibm.com (Linux
2.6.18-274.17.1.el5)
CorbaFrameworkBase Release ID: 10.1.0
CorbaFrameworkBase Jar Build Date: Wed May 08 2019 07:30:50 UTC on rhat5es-
build1.hursley.ibm.com (Linux 2.6.18-274.17.1.el5)
CorbaFrameworkTMF814 Release ID: 10.1.0
CorbaFrameworkTMF814 Jar Build Date: Wed May 08 2019 07:30:50 UTC 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: Thu Sep 20 2018 10:31:00 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_161, java runtime version:
8.0.5.10 - pxi3280sr5fp10-20180214_01(SR5 FP10), from this directory : /opt/ibm-java-
i386-80/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)
```

2 Example Configuration

2.1 Checking the object name

You can check the object name using the dumpns for non-SSL configurations with the naming service enabled:

```
cd $NCHOME/omnibus/probes/java/corba
./dumpns <emshost> 15001
```

Otherwise you can use the dior command to check the IOR string provided by the EMS administrator.

```
cd $NCHOME/omnibus/probes/java/corba/jacorb-3.3/bin
./dior -f /tmp/hu2000.ior
```

2.2 Checking the ORBLocalPort

If the ORBLocalPort is not able to be seen from the EMS server, events will be lost and the EMS will close the CORBA connection after a time out period.

You can can check the ports availability using curl and netcat.

Example probe property settings.

```
ORBLocalHost : '192.162.20.20'
ORBLocalPort : 12345
```

Create a netcat listener on the probe server on the ORBLocalPort whilst the probe is shutdown.

```
nc -l -p 12345
```

From the EMS use curl to connect to the listeners port.

```
curl telnet://192.162.20.20:12345
```

Use the same details that are set in the probe property file.

2.3 Property file

```
# Object Server connection
Server                  : 'AGG_P'
ServerBackup            : 'AGG_B'
# Best practice
NetworkTimeout          : 15
PollServer              : 60
# Buffering
Buffering               : 1
BufferSize              : 200
FlushBufferInterval     : 9
# Performance tuning
DisableDetails          : 1
#
NamingServiceHost       : '<emshost>'
NamingServicePort       : 15001
#
# HU2000 naming
NamingContextPath       :
'TMF_MTNM.Class/HUAWEI.Vendor/Huawei\\/U2000.EmsInstance/2\\.0.Version/Huawei\\/U2000.Ems
SessionFactory_I'
#
# Alternate naming
#NamingContextPath       :
'TMF_MTNM.Class/HUAWEI.Vendor/Huawei\/U2000.EmsInstance/2\.0.Version/Huawei/U2000.EmsSess
ionFactory_I'
# Synhronisation
InitialResync           : 'true'
#
# ORB Local settings
# Accessible from Huawei U2000 server
#
#ORBLocalHost            : '<probe-fqdn>'
#ORBLocalPort            : 12345
#
# Debugging
#
# MessageLevel            : 'debug'
# ORBDebug                : 'true'
# ORBDebugFile            : '$NCHOME/omnibus/log/U2000.EmsSessionFactory.orb.log'
#
# Heartbeating
HeartbeatInterval       : 60
#
# Command line interface
#
CommandPort             : 0
NHttpd.EnableHTTP       : TRUE
NHttpd.ListeningHostname : 'localhost'
NHttpd.ListeningPort    : 11111
NHttpd.AccessLog        : "$NCHOME/omnibus/log/U2000.11111.nhttpd.access.log"
# EOF
```

3 TLS/SSL Configuration

The example TLS/SSL configuration uses the default Huawei U2000 development certificates.

3.1 Creating the JKS keystore

Huawei U2000 EMS certificates:

server.cer
server_key.pem
trust.cer

3.1.1 Creating the CA certificate files from the servers trust.cer

The following shows how to create the two CA certificates from the Huawei U2000 servers trust.cer

File : trust.cer

```
-----BEGIN CERTIFICATE-----
##########################
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
##########################
-----END CERTIFICATE-----
```

Note: Holds two cetificates.

File : CA1.cer

```
-----BEGIN CERTIFICATE-----
##########################
-----END CERTIFICATE-----
```

File : CA2.cer

```
-----BEGIN CERTIFICATE-----
##########################
-----END CERTIFICATE-----
```

Example checks:

```
keytool -printcert -file CA1.cer
Owner:  CN=networkossCA, ...O=Huawei Technologies, ST=GuangDong, C=CN
Issuer: CN=huaweiossCA, ...O=Huawei Technologies, ST=GuangDong, C=CN


keytool -printcert -file CA2.cer
Owner:  CN=huaweiossCA, ...O=Huawei Technologies, ST=GuangDong, C=CN
Issuer: CN=huaweiossCA, ...O=Huawei Technologies, ST=GuangDong, C=CN
```

3.1.2 Creating a full chain PFX [PKCS12] file

The following shows how to create the certificate.pfx file from the Huawei servers certificates.

```
cp server.cer fullchain_server.cer
cat CA1.cer >> fullchain_server.cer
cat CA2.cer >> fullchain_server.cer
```

Creating the PFX file from fullchain_server.cer and server_key.pem files.

```
openssl pkcs12 -export -out certificate.pfx -inkey server_key.pem -in
fullchain_server.cer -certfile trust.cer -name server
Enter pass phrase for server_key.pem:
Enter Export Password:
Verifying - Enter Export Password:

Using the default password : Changeme_123
```

To check the PFX file:

```
keytool -list -keystore certificate.pfx -storepass Changeme_123
```

3.1.3 Creating the JKS keystore for probe use : server.jks

```
keytool -importkeystore -srckeystore certificate.pfx -srcstoretype pkcs12
-destkeystore server.jks -deststoretype JKS

keytool -list -keystore server.jks -storepass Changeme_123

Your keystore contains 1 entries
server, Apr 19, 2021, keyEntry,

keytool -keystore server.jks -alias ca1 -storepass Changeme_123 -import -file CA1.cer
keytool -keystore server.jks -alias ca2 -storepass Changeme_123 -import -file CA2.cer

keytool -list -keystore server.jks -storepass Changeme_123

Your keystore contains 3 entries
ca2, Apr 19, 2021, trustedCertEntry,
ca1, Apr 19, 2021, trustedCertEntry,
server, Apr 19, 2021, keyEntry,
```

3.2 Setting the probe properties

File : $NCHOME/omnibus/probes/linux2x86/huawei_u2000_corba.props

```
# SSL Configuration
#
# keytool -list -keystore $NCHOME/omnibus/probes/JKS/HU2000/server.jks -storepass
Changeme_123
# Your keystore contains 3 entries
# ca2, Apr 19, 2021, trustedCertEntry,
# ca1, Apr 19, 2021, trustedCertEntry,
# server, Apr 19, 2021, keyEntry,
#
EnableSSL                       : 'true'
KeyStore                        : '$NCHOME/omnibus/probes/JKS/HU2000/server.jks'
KeyStorePassword                : 'Changeme_123'
SecurityProtocol                : 'TLSv1.2'
```

File : $NCHOME/omnibus/probes/java/nco_p_huawei_u2000_corba.env

```
# Bidirectional handshakes
NCO_JPROBE_JAVA_FLAGS="-Djacorb.security.ssl.client.supported_options=60
$NCO_JPROBE_JAVA_FLAGS"
NCO_JPROBE_JAVA_FLAGS="-Djacorb.security.ssl.client.required_options=60
$NCO_JPROBE_JAVA_FLAGS"
NCO_JPROBE_JAVA_FLAGS="-Djacorb.security.ssl.server.supported_options=60
$NCO_JPROBE_JAVA_FLAGS"
NCO_JPROBE_JAVA_FLAGS="-Djacorb.security.ssl.server.required_options=60
$NCO_JPROBE_JAVA_FLAGS"
echo "NCO_JPROBE_JAVA_FLAGS=$NCO_JPROBE_JAVA_FLAGS"


# Trust - overrides Java's cacerts
NCO_JPROBE_JAVA_FLAGS="-
Djavax.net.ssl.trustStore=$NCHOME/omnibus/probes/JKS/HU2000/server.jks
-Djavax.net.ssl.trustStorePassword=Changeme_123 $NCO_JPROBE_JAVA_FLAGS"

# Debugging and extra customisation

# Keystore - defined in probe property file
#NCO_JPROBE_JAVA_FLAGS="-
Djavax.net.ssl.keyStore=$NCHOME/omnibus/probes/JKS/HU2000/server.jks
-Djavax.net.ssl.keyStorePassword=Changeme_123 $NCO_JPROBE_JAVA_FLAGS"
# TLS  - defined in probe property file
#NCO_JPROBE_JAVA_FLAGS="-Djdk.tls.client.protocols=TLSv1.2 -Dhttps.protocols=TLSv1.2
$NCO_JPROBE_JAVA_FLAGS"

# SSL debug logging for start-up issues
# NCO_JPROBE_JAVA_FLAGS="-Djavax.net.debug=all:handshake:verbose
$NCO_JPROBE_JAVA_FLAGS"

# Operating systems Java - JAVA_HOME
# NCO_PROBE_JRE=/usr
```

3.3 Debugging SSL

To debug the SSL handshakes enable handshake or all debug logging. Remember to disable this extra debug logging after resolving any issues, as it may cause unxpected problems. Run the probe from the command line when debugging SSL issues incase any important messages are sent to standard output.

File $NCHOME/omnibus/probes/java/nco_p_huawei_u2000_corba.env

```
# Debugging
# Create a unique log file name based on probe
PROBENAME=`echo $PROGRAM | awk -Fnco_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
```