Tivoli Netcool Support's Guide to The email probe

by
Jim Hutchinson

Document release: 2.4

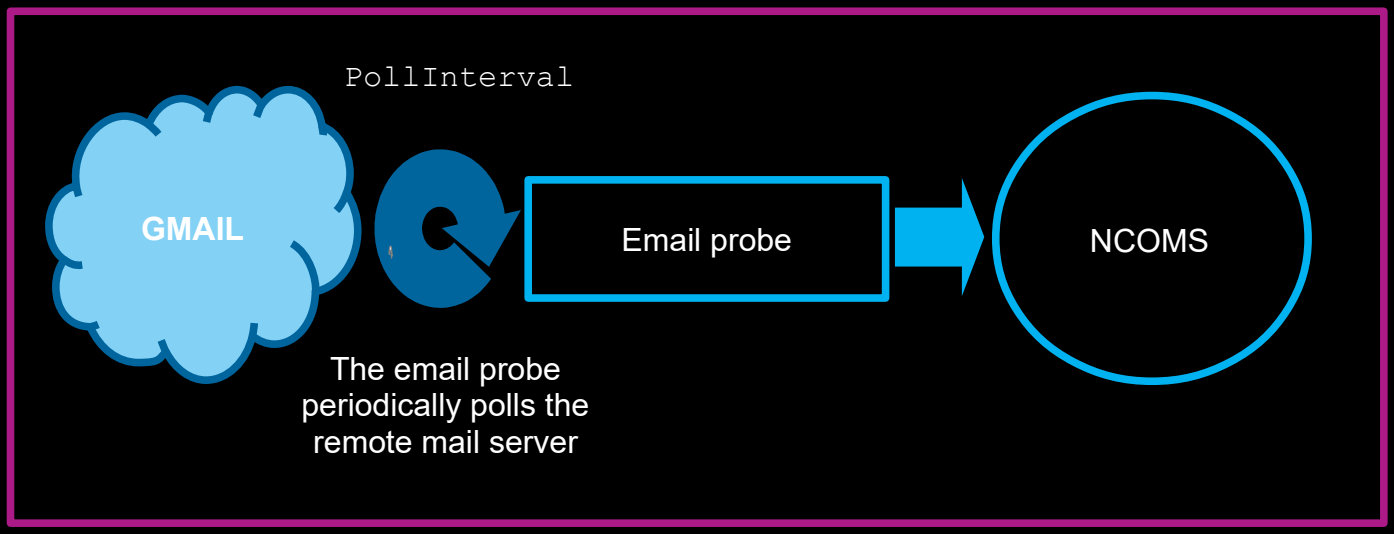
Table of Contents

1 Introduction

Th email probe is a non-native probe designed to collect mails from a mail server, much like a normal mail client. It requires a number of property settings to work correctly and uses nco_g_crypt to encrypt passwords by default. These days the email probe usually connects to the mail server using SSL which can complicate the probes configuration, as the required version of Java needs to be used, as well as the correct SSL certificates.

1.1 Overview

The email probe works by connecting to the remote mail server periodically using a username and password. Provided the correct security details are provided, the probe will read the contents of the users mailbox and create events based on the mail files read that match the probes filter, the email.rules file and generated tokens.



2 Windows Configuration

2.1 Debugging on Windows

Debugging the email probe on Windows is a little bit more complicated than UNIX.
Note that debugging should be disabled once the problem is resolved, and the probe is placed in a production environment.

2.1.1 Non-native debugging

Create a backup of the nco_p_email.bat file and then add the following lines:

e.g.

```
REM *** Set Debugging
set NDE_DEFAULT_LOG_LEVEL=debug
set NCO_P_NONNATIVE_TRANSCRIPT=%NCHOME%\omnibus\log\email_nonnative_debug.log
set NDE_FORCE_LOG_MODULE=%NCHOME%\omnibus\log\email_nonnative_forced_debug.log
```

If you are debugging other non-native probes, use a unique prefix for the non-native log file names.

2.1.2 SSL debugging

In order to debug SSL connections on Windows you need to add the SSL handshake argument to the Java command line options, which are available in the nco_p_email.bat:

e.g.

```
REM *** args to execute the probe
set PROGARGS=javaw -Djavax.net.debug=ssl:handshake:verbose -Xrs -cp %PROBE_CLASSPATH
%;%NSPROBE_CLASSPATH%;%MAIL_CLASSPATH% nco_p_email
```

Alongside this, the non-native debug logging needs to be enabled.

When the probe is run it will log message relating to the SSL Stores and certificates.

You can use grep to find what is going on, use a log analyser tool, or else review the files manually.

e.g.

```
grep '\*\*\*' nonnative_debug.log
*** ClientHello, TLSv1.2
*** ServerHello, TLSv1.2
*** Certificate chain
*** ECDH ServerKeyExchange
*** ServerHelloDone
*** ECDHClientKeyExchange
*** Finished
*** Finished
***
```

2.2 Windows GMAIL configuration

2.2.1 The GMAIL account

For the purposes of testing the email probes configuration you can use the GMAIL service, which supports SSL IMAPS connectivity. With the following GMAIL settings, the email probe was configured:

```
Username          : myemailprobe@gmail.com
Password          : netcool
Security Setting :  Allow third party Apps
IMAP Setting      :  Forwarding and POP/IMAP → enable IMAP Access
```

GMAIL now requires two-step authentication and App-passwords.

2.2.2 Java Configuration

Because GMAIL requires TLS v1.2 the latest version of Java was required to be configured. This is done by modifying the nco_p_email.bat file to include the required %JAVE_HOME% in the probes %PATH%.

e.g.

```
REM *** Set JAVA_HOME
set JAVA_HOME="C:\Program Files (x86)\Java\jre1.8.0_77"
set PATH="%JAVA_HOME%\bin";%PATH%
```

2.2.3 KeyStore file creation

In order to create Java Store, you need to set the Java path to the same path the probe is going to use, and then add the required SSL certificates.

e.g.

```
cd C:\Program Files (x86)\Java\jre1.8.0_77
.\keytool.exe -importcert -file "c:\temp\GeoTrustGlobalCA.pem" -keystore
"c:\temp\MOZILLA.JKS"

.\keytool.exe -list -keystore "c:\temp\MOZILLA.JKS"
Enter keystore password:

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 1 entry

mykey, 04-Jul-2016, trustedCertEntry,
Certificate fingerprint (SHA1): ...
```

2.2.4 Probe Property file

The GMAIL support site discusses the correct server to use, and how to configure the client connection. For SSL the email probe requires the IMAPS setting and the SSL port 993 to be set. The default AuthType setting is userpass, which is correct for IMAP, so does not need to be set.

```
MessageLevel                    : "debug"
Name                            : "myemailprobe"
RulesFile                       : "$OMNIHOME/probes/win32/email.rules"
Server                          : "MYCOMS"
NetworkTimeout                  : 15
PollServer                      : 30
# Mail settings
Protocol                        : "imaps"
Port                            : 993
DeleteEmails                    : "true"
Hostname                        : "imap.gmail.com"
Username                        : "myemailprobe@gmail.com"
Password                        : "ECEDBJAGBJFHGD"
PollInterval                    : 60
EmailSocketTimeOut              : 11
Filter                          : "#TEST"
# Buffering
Buffering                       : 1
BufferSize                      : 200
FlushBufferInterval             : 11
# SSL configuration
KeyStoreFile                    :
"C:\\IBM\\Tivoli\\Netcool\\omnibus\\probes\\win32\\MOZILLA.JKS"
KeyStorePassword                : "ECEDBJAGBJFHGD"
TrustStoreFile                  :
"C:\\IBM\\Tivoli\\Netcool\\omnibus\\probes\\win32\\MOZILLA.JKS"
TrustStorePassword              : "ECEDBJAGBJFHGD"

# EOF
```

2.3 Example files

2.3.1 SSL certificate

You can obtain SSL CA certificates from your desktops certificate store for use with the email probe with GMAIL.

This is an example of the GeoTrust Global CA certificate, which is held in the GeoTrustGlobalCA.pem file, for the example, and added to a new Jave KeyStore:

```
-----BEGIN CERTIFICATE-----
MIIDVDCCAjygAwIBAgIDAjRWMA0GCSqGSIb3DQEBBQUAMEIxCzAJBgNVBAYTAlVT
MRYwFAYDVQQKEw1HZW9UcnVzdCBJbmMuMRswGQYDVQQDExJHZW9UcnVzdCBHbG9i
YWwgQ0EwHhcNMDIwNTIxMDQwMDAwWhcNMjIwNTIxMDQwMDAwWjBCMQswCQYDVQQG
EwJVUzEWMBQGA1UEChMNR2VvVHJ1c3QgSW5jLjEbMBkGA1UEAxMSR2VvVHJ1c3Qg
R2xvYmFsIENBMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2swYYzD9
9BcjGlZ+W988bDjkcbd4kdS8odhM+KhDtgPpTSEHCIjaWC9mOSm9BXiLnTjoBbdq
fnGk5sRgprDvgOSJKA+eJdbtg/OtppHHmMlCGDUUna2YRpIuT8rxh0PBFpVXLVDv
iS2Aelet8u5fa9IAjbkU+BQVNdnARqN7csiRv8lVK83Qlz6cJmTM386DGXHKTubU
1XupGc1V3sjs0l44U+VcT4wt/lAjNvxm5suOpDkZALeVAjmRCw7+OC7RHQWa9k0+
bw8HHa8sHo9gOeL6NlMTOdReJivbPagUvTLrGAMoUgRx5aszPeE4uwc2hGKceeoW
MPRfwCvocWvk+QIDAQABo1MwUTAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBBTA
ephojYn7qwVkDBF9qn1luMrMTjAfBgNVHSMEGDAWgBTAephojYn7qwVkDBF9qn1l
uMrMTjANBgkqhkiG9w0BAQUFAAOCAQEANeMpauUvXVSOKVCUn5kaFOSPeCpilKIn
Z57QzxpeR+nBsqTP3UEaBU6bS+5Kb1VSsyShNwrrZHYqLizz/Tt1kL/6cdjHPTfS
tQWVYrmm3ok9Nns4d0iXrKYgjy6myQzCsplFAMfOEVEiIuCl6rYVSAlk6l5PdPcF
PseKUgzbFbS9bZvlxrFUaKnjaZC2mqUPuLk/IH2uSrW4nOQdtqvmlKXBx4Ot2/Un
hw4EbNX/3aBd7YdStysVAq45pmp06drE57xNNB6pXE0zX5IJL4hmXXeXxx12E6nV
5fEWCRE11azbJHFwLJhWC9kXtNHjUStedejV0NxPNO3CBWaAocvmMw==
-----END CERTIFICATE-----
```

Note: This certificate has expired.

2.3.2 Top of the nco_p_email.bat file

```
@echo off
setlocal

REM *** remove spaces from OMNIHOME variable or they screw things up
call :get_new_omnihome "%OMNIHOME%"

REM *** set up classpath variables for the Email probe
REM *** check first if we have a pre V7.1 Omnihome\probes directory structure
if exist "%OMNIHOME%\probes\nt351\nco_p_email.jar" (
           set CLASS_DIR=%NEWOMNIHOME%\probes\nt351
           )

if exist "%OMNIHOME%\probes\win32\nco_p_email.jar" (
           set CLASS_DIR=%NEWOMNIHOME%\probes\win32
           )

set PROBE_CLASSPATH=%CLASS_DIR%\nco_p_email.jar
set NSPROBE_CLASSPATH=%CLASS_DIR%\NSProbe.jar
set MAIL_CLASSPATH=%CLASS_DIR%\mail.jar;%CLASS_DIR%\activation.jar

REM *** Set JAVA_HOME
set JAVA_HOME="C:\Program Files (x86)\Java\jre1.8.0_77"
set PATH="%JAVA_HOME%\bin";%PATH%
REM *** Set Debugging
set NDE_DEFAULT_LOG_LEVEL=debug
set NCO_P_NONNATIVE_TRANSCRIPT=%NCHOME%\omnibus\log\nonnative_debug.log
set NDE_FORCE_LOG_MODULE=%NCHOME%\omnibus\log\nonnative_forced_debug.log

REM *** args to execute the probe
set PROGARGS=javaw -Djavax.net.debug=ssl:handshake:verbose -Xrs -cp %PROBE_CLASSPATH
%;%NSPROBE_CLASSPATH%;%MAIL_CLASSPATH% nco_p_email
```

2.3.3 Debug log file messages

The Following messages are seen when the probe successfully connects to the GMAIL server with SSL:

```
Debug: D-JPR-000-000: Enable javax.net.debug=ssl)
Debug: D-JPR-000-000: Connecting to mail server
Debug: D-JPR-000-000: Retrieving store object
Debug: D-JPR-000-000: Using TLS authentication
Debug: D-JPR-000-000: Enable SSL
Debug: D-JPR-000-000: Enable STARTTLS
Debug: D-JPR-000-000: Getting store object
Debug: D-JPR-000-000: Successfully retrieved store object
Debug: D-JPR-000-000: Connecting to mail server (Timeout in 11 secs if failing)
Debug: D-JPR-000-000: Successfully connected to mail server
Debug: D-JPR-000-000: Retrieving folder
Debug: D-JPR-000-000: Successfully retrieved folder
Debug: D-JPR-000-000: Opening folder
Debug: D-JPR-000-000: Successfully opened folder
Debug: D-JPR-000-000: Closing folder
Debug: D-JPR-000-000: Closing connection
```

3 UNIX [Linux] Configuration

The email probe configuration on linux is the same as Windows, although there are differences in the syntax and available tools.

3.1 Debugging on UNIX

3.1.1 Non-native debugging

The non-native debug logging allows full SSL debug logging.
Set the paths and log file names as required.

```
# ENABLE NON-NATIVE PROBE DEBUG LOGGING
# Uncomment to activate
#
# Create a unique log file name
# PROBENAME=`echo $PROGRAM | awk -Fnco_p_ '{print $2}'`
# UNIQUENAME=${PROBENAME}.$$
# export UNIQUENAME
# echo "UNIQUENAME=${UNIQUENAME}"
# NDE_DEFAULT_LOG_LEVEL="debug"
# NDE_FORCE_LOG_MODULE="$NCHOME/omnibus/log/${UNIQUENAME}_forced.log"
# NCO_P_NONNATIVE_TRANSCRIPT="$NCHOME/omnibus/log/${UNIQUENAME}_nonnative.log"
# export NDE_DEFAULT_LOG_LEVEL
# export NDE_FORCE_LOG_MODULE
# export NCO_P_NONNATIVE_TRANSCRIPT
```

3.1.2 SSL debugging

```
# Uncomment to activate one or other
# Debug SSL Handshakes
# NCO_JPROBE_JAVA_FLAGS="-Djavax.net.debug=ssl:handshake:verbose
$NCO_JPROBE_JAVA_FLAGS"
# Debug ALL SSL Handshakes
# NCO_JPROBE_JAVA_FLAGS="-Djavax.net.debug=all:handshake:verbose
$NCO_JPROBE_JAVA_FLAGS"
```

3.2 Microsoft Outlook example [IMAPS]

3.2.1 Outlook email account

```
Username          : testoutlook@outlook.com
Password          : netcool
IMAP Setting      : POPs/IMAPs enabled by default

IMAPS service
Server name: outlook.office365.com
Port: 993
Encryption method: TLS
```

3.2.2 Truststore

The openssl command shows the mail service being attached to a root CA in the default certificates file, found in the Java CACERTs file. Copy this file to a suitable location for reference by the probe and update the store password if required. The KeyStore property can use the same file.

3.2.3 Property file settings

```
# Best practice
NetworkTimeout      : 30
PollServer          : 60
# Buffering
Buffering           : 1
BufferSize          : 200
FlushBufferInterval: 9
# email settings
Protocol            : "imaps"
Port                : 993
DeleteEmails        : "false"
Hostname            : "outlook.office365.com"
Username            : "testoutlook@outlook.com"
Password            : "CLEHBDGEFPDLGLFCGN"
# Extra email settings
Filter              : '#netcool'
EmailSocketTimeOut  : 30
PollInterval        : 60
#
# openssl s_client -connect outlook.office365.com:993
# Owner: CN=outlook.com, O=Microsoft Corporation, L=Redmond, ST=Washington, C=US
# Issuer: CN=DigiCert Cloud Services CA-1, O=DigiCert Inc, C=US
#
# Server uses the standard CA available in java cacerts
#
# keytool -list -keystore cacerts.jks -storepass changeit
#
KeyStoreFile        :
'/opt/IBM/tivoli/netcool/omnibus/probes/linux2x86/OUTLOOK/cacerts.jks'
KeyStorePassword    : 'DHEGAGBBBBENGAGF'    #changeit
TrustStoreFile      :
'/opt/IBM/tivoli/netcool/omnibus/probes/linux2x86/OUTLOOK/cacerts.jks'
TrustStorePassword : 'DHEGAGBBBBENGAGF'   #changeit
```

3.3 Microsoft Outlook example [POP3S]

3.3.1 Outlook email account

```
Username          : testoutlook@outlook.com
Password          : netcool
IMAP Setting      : POPs/IMAPs enabled by default

POPs service
Server name: outlook.office365.com
Port: 995
Encryption method: TLS
```

3.3.2 Truststore

The openssl command shows the mail service being attached to a root CA in the default certificates file, found in the Java CACERTs file. Copy this file to a suitable location for reference by the probe and update the store password if required. The KeyStore property can use the same file.

3.3.3 Property file settings

```
# Best practice
NetworkTimeout      : 30
PollServer          : 60
# Buffering
Buffering           : 1
BufferSize          : 200
FlushBufferInterval: 9
# email settings
Protocol            : "pop3s"
Port                : 995
DeleteEmails        : "false"
Hostname            : "outlook.office365.com"
Username            : "testoutlook@outlook.com"
Password            : "CLEHBDGEFPDLGLFCGN"
# Extra email settings
Filter              : '#netcool'
EmailSocketTimeOut : 30
PollInterval        : 60
#
# openssl s_client -connect outlook.office365.com:995
# Owner: CN=outlook.com, O=Microsoft Corporation, L=Redmond, ST=Washington, C=US
# Issuer: CN=DigiCert Cloud Services CA-1, O=DigiCert Inc, C=US
#
# Server uses the standard CA available in java cacerts
#
# keytool -list -keystore cacerts.jks -storepass changeit
#
KeyStoreFile        :
'/opt/IBM/tivoli/netcool/omnibus/probes/linux2x86/OUTLOOK/cacerts.jks'
KeyStorePassword    : 'DHEGAGBBBBENGAGF'    #changeit
TrustStoreFile      :
'/opt/IBM/tivoli/netcool/omnibus/probes/linux2x86/OUTLOOK/cacerts.jks'
TrustStorePassword : 'DHEGAGBBBBENGAGF'   #changeit
```

3.4 GMAIL configuration

3.4.1 GMAIL account

```
Username         : myemailtestaccount@gmail.com
Password         : netcool
Security Setting :  Allow third party Apps [discontinued 2022]
IMAP Setting     :  Forwarding and POP/IMAP → enable IMAP Access
```

GMAIL now requires two-step authentication and App-passwords.
Refer to the Gmail settings and help for current configuration requirements.

3.5 Java Configuration

You can configure the probe using the probe to use a specific Java and set Java options using the NCO_JPROBE_JAVA_XFLAGS environment variable.

```
# Set Sun Java
# NCO_PROBE_JRE=/opt/sun-java-x86_64-80/jre
# Fix JVM memory usage
# NCO_JPROBE_JAVA_XFLAGS="-Xms64M -Xmx1024M $NCO_JPROBE_JAVA_FLAGS"
# Enforce TLSv1.2
# NCO_JPROBE_JAVA_FLAGS="-Dhttps.protocols=TLSv1.2 $NCO_JPROBE_JAVA_FLAGS"
```

3.6 Probe Property file

The following is an example probe property file, configured for connecting to the GMAIL account given earlier. Passwords are encrypted using nco_g_crypt but can be encrypted using the AES crypt method.

```
Server              : 'AGG_P'
ServerBackup        : 'AGG_B'
# Best practice
NetworkTimeout      : 30
PollServer          : 60
# Buffering
Buffering           : 1
BufferSize          : 200
FlushBufferInterval: 9
# email settings
Protocol            : "imaps"
Port                : 993
DeleteEmails        : "false"
Hostname            : "imap.gmail.com"
Username            : "myemailtestaccount@gmail.com"
Password            : "AABBCCDDEEFFGGHH"
# Extra email settings
Filter              : '#netcool'
EmailSocketTimeOut : 30
PollInterval        : 60
#
# keytool -list -keystore keystore.jks -storepass changeit
# imap.gmail.com, Jan 11, 2021, trustedCertEntry,
# geotrust, Jan 11, 2021, trustedCertEntry,
#
# For Gmail you can use cacerts from the current Java
#
# KeyStoreFile       : '/opt/IBM/tivoli/netcool/omnibus/probes/linux2x86/cacerts.jks'
#
# Example secured email server usage
#
KeyStoreFile        : '/opt/IBM/tivoli/netcool/omnibus/probes/linux2x86/keystore.jks'
KeyStorePassword    : 'DHEGAGBBBBENGAGF'    #changeit
TrustStoreFile      : '/opt/IBM/tivoli/netcool/omnibus/probes/linux2x86/cacerts.jks'
TrustStorePassword : 'DHEGAGBBBBENGAGF'   #changeit
#EOF
```

3.7 KeyStore file creation example

In order to create Java Store, you need to set the Java path to the same path the probe is going to use, and then add the required SSL certificate[s].
e.g.

```
find $NCHOME -name java
$NCHOME/platform/linux2x86/jre64_1.8.0/jre/bin/java
PATH=$NCHOME/platform/linux2x86/jre64_1.8.0/jre/bin:$PATH
which keytool
cd $NCHOME/omnibus/probes/linux2x86
keytool -printcert -file GeoTrustGlobalCA.pem
Owner: CN=GeoTrust Global CA, O=GeoTrust Inc., C=US
Issuer: CN=GeoTrust Global CA, O=GeoTrust Inc., C=US
```

Import the required CA for the email service into the probes keystore.

```
keytool -importcert -file /tmp/GeoTrustGlobalCA.pem -alias geotrustglobalca -keystore
./myemailtest2021.jks
Enter keystore password:
Re-enter new password:

keytool -list -keystore  ./myemailtest2021.jks -storepass changeit
Your keystore contains 1 entry
geotrustglobalca, Jan 29, 2021, trustedCertEntry,
```

The truststore can be taken from the Java installs cacerts file or else created from the required CA certificates.
e.g

```
find $NCHOME -name cacerts
$NCHOME/platform/linux2x86/jre64_1.8.0/jre/lib/security/cacerts
cp $NCHOME/platform/linux2x86/jre64_1.8.0/jre/lib/security/cacerts ./my_cacerts.jks
keytool -list -keystore  ./my_cacerts.jks -storepass changeit

geotrustuniversalca, Jul 15, 2011, trustedCertEntry,
geotrustprimaryca, Jul 15, 2011, trustedCertEntry,
geotrustglobalca, Jul 18, 2003, trustedCertEntry,
```

3.8 Example files

3.8.1 SSL certificate

You can obtain SSL CA certificates from the public CA certificate creators website.

This is an example of the GeoTrust Global CA certificate, which is held in a text file, and added to a new Java KeyStore:

```
-----BEGIN CERTIFICATE-----
MIIDVDCCAjygAwIBAgIDAjRWMA0GCSqGSIb3DQEBBQUAMEIxCzAJBgNVBAYTAlVT
MRYwFAYDVQQKEw1HZW9UcnVzdCBJbmMuMRswGQYDVQQDExJHZW9UcnVzdCBHbG9i
YWwgQ0EwHhcNMDIwNTIxMDQwMDAwWhcNMjIwNTIxMDQwMDAwWjBCMQswCQYDVQQG
EwJVUzEWMBQGA1UEChMNR2VvVHJ1c3QgSW5jLjEbMBkGA1UEAxMSR2VvVHJ1c3Qg
R2xvYmFsIENBMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2swYYzD9
9BcjGlZ+W988bDjkcbd4kdS8odhM+KhDtgPpTSEHCIjaWC9mOSm9BXiLnTjoBbdq
fnGk5sRgprDvgOSJKA+eJdbtg/OtppHHmMlCGDUUna2YRpIuT8rxh0PBFpVXLVDv
iS2Aelet8u5fa9IAjbkU+BQVNdnARqN7csiRv8lVK83Qlz6cJmTM386DGXHKTubU
1XupGc1V3sjs0l44U+VcT4wt/lAjNvxm5suOpDkZALeVAjmRCw7+OC7RHQWa9k0+
bw8HHa8sHo9gOeL6NlMTOdReJivbPagUvTLrGAMoUgRx5aszPeE4uwc2hGKceeoW
MPRfwCvocWvk+QIDAQABo1MwUTAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBBTA
ephojYn7qwVkDBF9qn1luMrMTjAfBgNVHSMEGDAWgBTAephojYn7qwVkDBF9qn1l
uMrMTjANBgkqhkiG9w0BAQUFAAOCAQEANeMpauUvXVSOKVCUn5kaFOSPeCpilKIn
Z57QzxpeR+nBsqTP3UEaBU6bS+5Kb1VSsyShNwrrZHYqLizz/Tt1kL/6cdjHPTfS
tQWVYrmm3ok9Nns4d0iXrKYgjy6myQzCsplFAMfOEVEiIuCl6rYVSAlk6l5PdPcF
PseKUgzbFbS9bZvlxrFUaKnjaZC2mqUPuLk/IH2uSrW4nOQdtqvmlKXBx4Ot2/Un
hw4EbNX/3aBd7YdStysVAq45pmp06drE57xNNB6pXE0zX5IJL4hmXXeXxx12E6nV
5fEWCRE11azbJHFwLJhWC9kXtNHjUStedejV0NxPNO3CBWaAocvmMw==
-----END CERTIFICATE-----
```

Note: This certificate has expired.

3.8.2 SSL Debug log file messages

Note: SSL Debug logging should be disabled after resolving issues with connectivity on start-up to prevent issues.

```
grep '\*\*\*' email_nonnative.log

*** ClientHello, TLSv1.2
***
*** ServerHello, TLSv1.2
***
*** Certificate chain
***
*** ECDH ServerKeyExchange
*** ServerHelloDone
*** Finished
*** Finished


grep '\*\*\*' email.MyEmailTest2021.log

02:25:58: Debug: D-UNK-105-000: *** Log Header Begin ***
02:25:58: Debug: D-UNK-105-000: *** Log Header End ***
02:25:59: Debug: D-JPR-105-000: ***
02:26:00: Debug: D-JPR-105-000: ***
02:26:00: Debug: D-JPR-105-000: ***
02:26:00: Debug: D-JPR-105-000: *** ServerHelloDone
02:26:00: Debug: D-JPR-105-000: *** ECDHClientKeyExchange
02:26:00: Debug: D-JPR-105-000: *** Finished
02:26:00: Debug: D-JPR-105-000: ***
02:26:00: Debug: D-JPR-105-000: *** Finished
02:26:00: Debug: D-JPR-105-000: ***
```

3.8.3 SSL issues

Missing trust store CA's

```
grep '\*\*\*' email_nonnative.log

*** ClientHello, TLSv1.2
***
*** ServerHello, TLSv1.2
***
*** Certificate chain

grep -i found email_nonnative.log
<nothing>
```

Further checking shows:

```
grep -i exception email_nonnative.log
main, handling exception: javax.net.ssl.SSLHandshakeException:
com.ibm.jsse2.util.h: PKIX path building failed:
java.security.cert.CertPathBuilderException: P
KIXCertPathBuilderImpl could not build a valid CertPath.; internal cause is:
        java.security.cert.CertPathValidatorException:
The certificate issued by CN=GlobalSign, O=GlobalSign, OU=GlobalSign Root CA - R2
is not trusted; internal cause is:
java.security.cert.CertPathValidatorException: Certificate chaining error
```

3.8.4 Information log file messages

06:57:40: Information: I-UNK-000-000: Connecting ...
06:57:40: Information: I-UNK-000-000: Using targets specified by properties
06:57:40: Information: I-UNK-000-000: 'AGG_P' is a primary server. Polling disabled.
06:57:40: Information: I-UNK-000-000: Probe registered with 'AGG_P'.
06:57:40: Information: I-UNK-000-000: Probewatch: Running ...
06:57:43: Information: I-UNK-000-000: Probewatch: Connected to mail server
06:58:46: Information: I-UNK-000-000: Probewatch: Connected to mail server
06:59:48: Information: I-UNK-000-000: Probewatch: Connected to mail server
07:00:52: Information: I-UNK-000-000: Probewatch: Connected to mail server
07:01:54: Information: I-UNK-000-000: Probewatch: Connected to mail server

3.9 Updating mail.jar

The mail.jar file usd by the email probes can be updated to make sure all of the known issues with the mail java classes are resolved.

Download the javax.mail.jar from a suitable source:

https://github.com/javaee/javamail/releases

Deploy the javax.mail.jar in the probes java directory.

For example.

```
cd $NCHOME/omnibus/probes/java/
mkdir  javamail_1.6.1
cp /tmp/javax.mail.jar .
```

Update the email probes environment with the new mail jar file.

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

```
#! /bin/ksh
#
# Replace the mail.jar with the javax.mail.jar
#
CLASSPATH=$NCHOME/omnibus/probes/java/javamail_1.6.1/javax.mail.jar
export CLASSPATH
#
# Set Sun Java
# NCO_PROBE_JRE=/opt/sun-java-x86_64-80/jre
#
# Debug message
echo “Using the JAVAX.MAIL.JAR”
echo "NCO_JPROBE_JAVA_FLAGS=$NCO_JPROBE_JAVA_FLAGS"
# EOF
```

3.10 Creating a Microsoft 365 truststore

The Microsft CA certificates are provided as p7b files. These need to be converted to PEM format and added to a JKS file for use with the email probe.

Example files from Microsoft's web site.
m365_intermediate_certs_20201013.p7b
m365_root_certs_20201012.p7b

Check the microsoft 365 servers certificate for the required service [IMAPS] matches the certificates in the files:

```
openssl s_client -connect outlook.office365.com:993
```

3.10.1 Extracting the PEM certificates

Check the p7b files contents:

```
openssl pkcs7 -print_certs -in m365_intermediate_certs_20201013.p7b
openssl pkcs7 -print_certs -in form der -inm365_root_certs_20201012.p7b
```

Convert the p7b file into a PEM file:

```
openssl pkcs7 -print_certs -in m365_intermediate_certs_20201013.p7b | grep -v subject |
grep -v issuer | grep . > m365_intermediate_certs.pem
openssl pkcs7 -print_certs -inform der -in m365_root_certs_20201012.p7b | grep -v subject
| grep -v issuer | grep . > m365_root_certs.pem
```

3.10.2 Creating the JKS trust store file

The example script allows PEM certificates to imported into a JKS trust store file from a single file, such as the ones created from the p7b files.

```
import_all_pems.sh
Usage: import_all_pems.sh [PEM_FILE] [KEYTOOL] [PASSWORD] [KEYSTORE]
```

The script requires the probes Java keytool to be provided, along with a suitable JKS store password and name.

Find the keytool path using:

```
find $NCHOME -name keytool
```

Remove the old JKS store file, beforehand, if the store is being recreated.

```
rm microsoft365.jks
```

Run the import_all_pems.sh script for each PEM file that needs to be added to the JKS store file.

```
./import_all_pems.sh m365_root_certs.pem
$NCHOME/platform/linux2x86/jre64_1.8.0/jre/bin/keytool netcool microsoft365.jks
./import_all_pems.sh m365_intermediate_certs.pem
$NCHOME/platform/linux2x86/jre64_1.8.0/jre/bin/keytool netcool microsoft365.jks
./import_all_pems.sh office365.cert
$NCHOME/platform/linux2x86/jre64_1.8.0/jre/bin/keytool netcool microsoft365.jks
```

Check the alias's in the JKS store file using:

```
keytool -list -keystore microsoft365.jks -storepass netcool | grep -v finger
```

Check the Owner and Issuer names in the JKS store file using:

```
keytool -v -list -keystore microsoft365.jks -storepass netcool | grep 'Owner:\|
Issuer:'
```

3.10.3 Example Script to import PEM Certificates

File : $NCHOME/omnibus/utils/import_all_pems.sh

```
#! /bin/sh
#
# Use the script and cer file from the openssl to create a JKS store file:
#
#For example:
#
# ./import_all_pems.sh all.pem $NCHOME/platform/linux2x86/jre64_1.8.0/jre/bin/keytool
netcool all.jks
#
# Which will create a JKS file called all.jks with password netcool
# using the multiple certificate file all.cer
#
if [ $# -ne 4 ]
then
echo "Usage: `basename $0` [PEM_FILE] [KEYTOOL] [PASSWORD] [KEYSTORE]"
exit
fi

PEM_FILE=$1
KEYTOOL=$2
PASSWORD=$3
KEYSTORE=$4

export PEM_FILE KEYTOOL PASSWORD KEYSTORE
export FILENAME COUNT ALIAS

# Count the number of certs in the PEM file
CERTS=$(grep 'END CERTIFICATE' $PEM_FILE| wc -l)
echo "CERTS=$CERTS"
# For every certificate in the PEM file,
# extract it and import into the JKS keystore
for COUNT in $(seq 0 $(($CERTS - 1)))
do
  FILENAME=`basename ${PEM_FILE}`
  ALIAS="${FILENAME}_$COUNT"
  echo "Creating $ALIAS"
  cat $PEM_FILE |
    awk "n==$COUNT {print}; /END CERTIFICATE/ {n++}" |
    $KEYTOOL -noprompt -import -trustcacerts \
            -alias $ALIAS -keystore $KEYSTORE -storepass $PASSWORD
done

if [ -f $KEYSTORE ]
then
echo "keytool -list -keystore $KEYSTORE -storepass $PASSWORD"
keytool -list -keystore $KEYSTORE -storepass $PASSWORD
else
echo " ERROR: Store file was not created : $KEYSTORE"
fi

# EOF
```

4 Appendix

4.1 Latest fixes

The email probe release 5.11 was released in June 2023.

```
PATCH probe-nco-p-email-5
REVISION 11
Known Issue DT202895 - Probe may times out its connection with mail server when
reading large emails. This issue has been fixed with de-coupling of probe connection
manager and message parser into seperate threads.
Temporarily disable Solaris build until future release.

REVISION 10
APAR IJ33729 - Probe may not run in Windows because the batch script used system JRE
to run 'javaw'. This issue has been fixed by making the batch script to run
executables from default JRE with OMNIbus installation.

REVISION 9
 APAR IJ27933 - Probe may stop reading email when issue arises within API during
accessing INBOX folder. This issue is fixed by probe creating separate thread to
connect and read emails, while imposing timeout based on EmailSocketTimeOut setting
for both purposes.

REVISION 8
APAR IJ28204 - Introduced new property ParseTextAttachment to make probe parse text
based attachments as part of email body and revert IJ20126 changes. Default is
'false'.

REVISION 7
APAR IJ26872 - Added new properties MaxBodyLength, MaxHeaderLength, MaxLineCount and
MaxLineSize to allow user configurable length or limit for element Body, Body_nn,
Header and Header_nn. Value of 0 for these new properties will disable length or
limit for these elements.

REVISION 6
APAR IJ20126
Fixed issue of text based email attachment mix up with email body.
- Fixed issue of inline attachments with empty ContentID of email MIME parts produced
by some email clients, which prevented  text based content of the inline attachments
from being recognized as part of email body.
- Limit length of header and body element to 1024 to avoid extremely long header and
body crashing non-native process.
Truncated elements will have a "(truncated)" at the end of text.
- Limit single line version of header and body elements to maximum of 4096 (eg.
Body_1 to Body_4096) to avoid delay in processing extremely large emails.
However, the total single lines count (eg. Body_LineCount) will still be obtained and
reflecting the actual total body lines.

RFE 137470   - Make attachment filenames available as token $Attachments,
$Attachment_1, $Attachment_2, ... $Attachment_Count.

REVISION 5
APAR IV92893 - The probe fails to parse emails that are missing a sent date or from
address.
```

```
REVISION 4
APAR IV67967 - Probe does not reset retry wait period when it successfully
reconnects.

REVISION 3
APAR IV23019 - Email Probe crashes when it reads the unsupported encoding email.

REVISION 2
APAR IV15487 - Turn off auto SSL debugging when MessageLevel property set to "Debug".

REVISION 1
Port in APAR IV01896, APAR IZ99337 and APAR IZ98976

REVISION 0
General maintenance release - incorporates all previous fixes since GA release 4_0
APAR IZ96878 -
Probe now will only try to match Body field of the mail message
with Filter property when FilterField property is empty or set to "body" (case
ignored).
```

4.2 Timeout settings

The timeout settings are Java options for specific email protocols, which can be set in the probes environment file.

```
#
# POP3
#
# mail.pop3.connectiontimeout
# Socket connection timeout value in milliseconds.
# This timeout is implemented by java.net.Socket. Default is infinite timeout.
#
# mail.pop3.timeout
# Socket read timeout value in milliseconds.
# This timeout is implemented by java.net.Socket. Default is infinite timeout.
#
# mail.pop3.writetimeout
# Socket write timeout value in milliseconds.
# This timeout is implemented by using a
java.util.concurrent.ScheduledExecutorService
# per connection that schedules a thread to close the socket if the timeout expires.
# Thus, the overhead of using this timeout is one thread per connection.
# Default is infinite timeout.
#
# NCO_JPROBE_JAVA_FLAGS="-Dmail.pop3.connectiontimeout=30000 $NCO_JPROBE_JAVA_FLAGS"
# NCO_JPROBE_JAVA_FLAGS="-Dmail.pop3.timeout=60000 $NCO_JPROBE_JAVA_FLAGS"
# NCO_JPROBE_JAVA_FLAGS="-Dmail.pop3.writetimeout=60000 $NCO_JPROBE_JAVA_FLAGS"
#
# SMTP:
#
# mail.smtp.connectiontimeout
# Socket connection timeout value in milliseconds.
# This timeout is implemented by java.net.Socket.
# Default is infinite timeout.
#
# mail.smtp.timeout
# Socket read timeout value in milliseconds.
# This timeout is implemented by java.net.Socket. Default is infinite timeout.
#
# mail.smtp.writetimeout
# Socket write timeout value in milliseconds.
# This timeout is implemented by using a
java.util.concurrent.ScheduledExecutorService
# per connection that schedules a thread to close the socket if the timeout expires.
# Thus, the overhead of using this timeout is one thread per connection.
# Default is infinite timeout.
#
# NCO_JPROBE_JAVA_FLAGS="-Dmail.smtp.connectiontimeout=30000 $NCO_JPROBE_JAVA_FLAGS"
# NCO_JPROBE_JAVA_FLAGS="-Dmail.smtp.timeout=60000 $NCO_JPROBE_JAVA_FLAGS"
# NCO_JPROBE_JAVA_FLAGS="-Dmail.smtp.writetimeout=60000 $NCO_JPROBE_JAVA_FLAGS"
```

4.3 SSL protocol settings

Enforcing protocols using the probes environment file and the Java option.

TLS protocols are TLSv1,TLSv1.1,TLSv1.2

For SMTP:
```
-Dmail.smtps.ssl.protocols=TLSv1.2
```

For IMAP:
```
-Dmail.imaps.ssl.protocols=TLSv1.2
```

For POP3:
```
-Dmail.pop3s.ssl.protocols=TLSv1.2
```

For example, to enforce TLSv1.2 for IMAPS use.

```
NCO_JPROBE_JAVA_FLAGS="-Dmail.imaps.ssl.protocols=TLSv1.2 $NCO_JPROBE_JAVA_FLAGS"
```

4.4 Checking services

4.4.1 Curl

The curl command can be used to check access to mail server services.

```
SMTP:
curl -v telnet://outlook.office365.com:25

POP3:
curl -v telnet://outlook.office365.com:110

POP3S:
curl -v --insecure telnet://outlook.office365.com:995

IMAP:
curl -v  --url "imap://outlook.office365.com/" -n

IMAPS:
curl -v --insecure --url "imaps://outlook.office365.com/" -n
```

4.4.2 OpenSSL

You can use openssl to check IMAPS access from the probe server.

```
openssl s_client -crlf -connect mailserver.company.com:993
CONNECTED(00000003)
...
Certificate chain
...
Server certificate
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
...
New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES256-GCM-SHA384
...
SSL-Session:
    Protocol  : TLSv1.2
    Cipher    : ECDHE-RSA-AES256-GCM-SHA384
...
---
* OK KPN102 Exchange IMAP4 service is ready.
? login domain\username password
? OK LOGIN completed.
? list "" "*"
...
? status inbox (messages)
* STATUS INBOX (messages ######)
? OK STATUS completed.
? logout
* BYE Microsoft Exchange Server IMAP4 server signing off.
? OK LOGOUT completed.
```