



NovaMap® TLS Plugin for Open Integration Engine v1.0.0: User Guide

Legal Notice

Although we exercise great care in creating this publication, NovaMap Health Limited assumes no responsibility for errors or omissions that may appear in this publication and reserves the right to change this publication at any time without notice.

© NovaMap Health Limited 2026. All Rights Reserved.

Our registered trademarks can be found at <https://www.novamap.health/>

All other names and marks are the property of their respective owners. In particular, please note that:

- This project is an independent, third-party work and is not affiliated with, endorsed by, sponsored by, or associated with NextGen Healthcare in any way.
- Open Integration Engine™, OIE™, and the OIE Logo are trademarks of the Open Integration Engine project

Contents

1. Overview	4
2. Installation	5
3. Launcher Compatibility	6
4. TLS Manager: Introduction	7
Login.....	7
Panel Display.....	9
Certificate Details.....	10
Verify Certificate.....	13
5. TLS Manager: Importing	14
Additional Trusted Certificates.....	14
Import Method 1: Choose Certificate File.....	15
Import Method 2: Paste Certificate.....	19
Import Method 3: Import From URL.....	21
Local Key Pairs.....	24
Import Key Pair.....	25
Show Private Keys.....	28
6. TLS Manager: Common Tasks	29
Certificate Rename.....	29
Expiry Warning.....	30
Certificate Replacement.....	31
Certificate Removal.....	32
7. Sender TLS Settings	33
Exclusive to TCP Sender in Server Mode.....	38
8. Listener TLS Settings	39
Exclusive to TCP Listener in Client Mode.....	44
Validation.....	45
9. API	46
URL.....	46
Access.....	46
API Calls.....	46
10. Known Issues	49
A. Deselection of mTLS in Client Mode of TCP Listener and TCP Sender.....	49
B. Certificate Alias and Upper Case.....	49
Acknowledgements	50
Appendix A	51
mirth.properties.....	51
Appendix B	52
Environment Variables.....	52
Appendix C	53
Third Party Java Libraries.....	53
Appendix D	54
IETF TLS Protocols.....	54

1. Overview

The NovaMap TLS plugin for Open Integration Engine (henceforth “TLS Plugin”) provides:

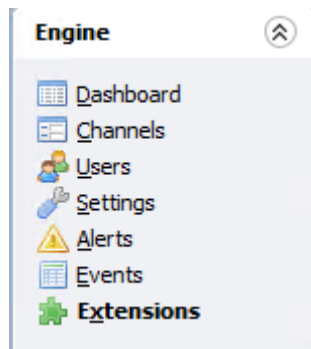
- a) Web based administration portal for certificate management.
- b) A REST API for certificate management, documented according to the OpenAPI specification.
- c) TLS and mTLS capability in HTTP, TCP and Web Services connector types.

2. Installation

The plugin is available to download from the NovaMap github repository:

<https://github.com/NovaMap-Health/tls-manager-plugin/releases>

To Install log in to Open Integration Engine and navigate to **Extensions**:



The installed Connectors and Plugins will be listed. At the bottom of the Extensions view is the **Install Extension from File System** section.

Click **Browse** and select your TLS plugin zip file.

Install Extension from File System

File:

Click **Install**. OIE will display the following notice:

The Open Integration Engine Server and Administrator must be restarted before your changes will take effect.

Installation is completed by restarting the OIE service/daemon.

3. Launcher Compatibility

The plugin has been extensively tested with the NextGen Mirth Connect Administrator Launcher.

The plugin is signed by CN=Novamap Health Ltd, O=Novamap Health Ltd, L=Cambridge, C=GB.

Depending on your preferred launcher you may be prompted to trust the certificate on first use.

4. TLS Manager: Introduction

Login

Certificate management for the TLS Plugin is done using a web-based user interface available at:

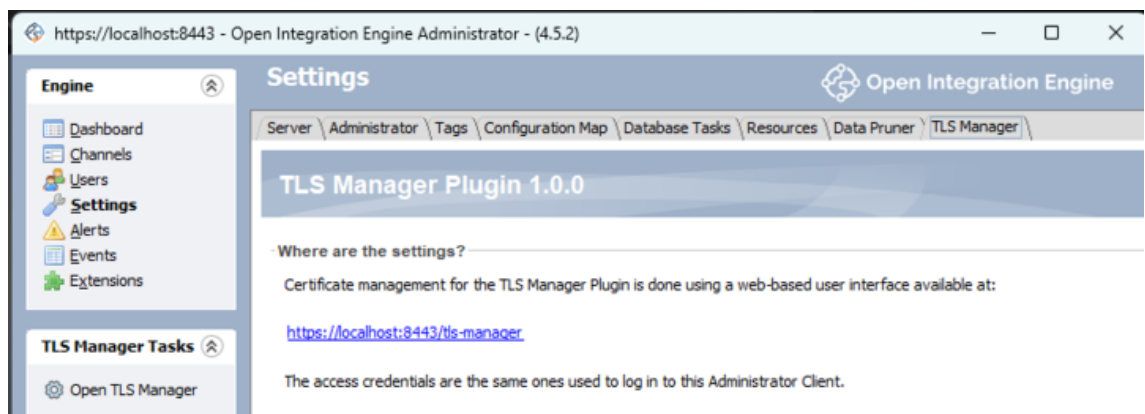
`https://<hostname>:port/tls-manager`

For example: `https://oiehost.example.com:8443/tls-manager`

Or locally: `https://localhost:8443/tls-manager`

Access credentials are the same ones used to log in to the OIE Administrator Client

The web page can also be accessed from the OIE Administrator Client by going to Settings and selecting the TLS Manager tab:



Categories

The TLS Manager divides management of TLS certificates into three areas.

The built-in Native Java Certificate Store, Additional Trusted Certificates and Local Key Pairs.


Category	Description	Import Options	Certificate Options
Native Java Certificate Store	Java certificate store.	<ul style="list-style-type: none"> None 	<ul style="list-style-type: none"> View Details
Additional Trusted Certificates	Certificates of remote clients or remote servers.	<ul style="list-style-type: none"> Import Certificates Import From URL 	<ul style="list-style-type: none"> View Details Edit Alias Remove
Local Key Pairs	Certificates with Private Keys. Typically representing the OIE instance owner.	<ul style="list-style-type: none"> Import Key Pair 	

Panel Display

In each category certificates are shown in a panel view that shows these data items:

- Alias
- Status
- Type
- Subject
- Issuer
- Valid From
- Valid To
- Fingerprint (SHA-1)

Example of the panel display featuring a certificate in Java's native store:


 **cn_digicert_assured_id_root_g2,ou_wwwdigicertcom,o_digicert_inc,c_us [jdk]** Valid
Root CA

Subject:
/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=DigiCert Assured ID Root G2

Issuer:
/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=DigiCert Assured ID Root G2

Valid From: Valid To:
2013-08-01 2038-01-15

Fingerprint (SHA-1):
A14B48D943EE0A0E40904F3CE0A4C09193515D3F

 [View Details](#)

Certificate Details

The View Details button displays the full details of a certificate:

- Status
- Alias
- Type
- Store
- Has Private Key
- Subject
- DNS Names
- IP Addresses
- Issuer
- Valid From
- Valid To
- SHA-1
- basicConstraints
- keyUsage
- extKeyUsage (if present)
- Channels in Use (if applicable)
- Raw Certificate (Base64)
- Private Key (Base64) (if applicable)
- Certificate Verification

Certificate Details example:

Certificate Details Valid

Basic Information

Alias	cn_digicert_assured_id_root_g2,ou_wwwdigicertcom,o_digicert_inc,c_us [jdk]	Type	Root CA
Store	Has Private Key		
Native	No		

Subject

/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=DigiCert Assured ID Root G2

Issuer

/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=DigiCert Assured ID Root G2

Validity Period

Valid From	Valid To
2013-08-01	2038-01-15

Certificate Details example continued:

Fingerprint

SHA-1

A14B48D943EE0A0E40904F3CE0A4C09193515D3F

Extensions

basicConstraints Critical

keyUsage Critical

digitalSignature, keyCertSign, cRLSign

Raw Certificate (Base64)

```
-----BEGIN CERTIFICATE----- MIID1jCCAn6gAwIBAgIQC5Mc0tY5Z+pnI7/Dr5r0SzANBgkqhkiG9w0BAQsFAD
B1 MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRG1naUN1cnQgSW5jMRkwFwYDVQQLExB3 d3cuZG1naW1naUN1cnQy29tMS
QwIgwYDVQQDExEaWdpQ2VydCBBC3N1cmVkaE1E1E1FJv b3QgRzIwHhcNMTMwODAxMTIwMDAwMzE1MTIwMDAw
wWJzB1MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRG1naUN1cnQgSW5jMRkwFwYDVQQLExB3d3cuZG1naW1naUN1 cnQuY29
tMSQwIgwYDVQQDExEaWdpQ2VydCBBC3N1cmVkaE1E1E1FJvb3QgRzIwHhcNMTMwODAxMTIwMDAwMzE1MTIwMDAw
AoIQAQDZ5ygvUj82ckmIkzTz+GoeMVSA n61UQbVH35ao1K+ALbkKz3X9iaV9JPrjIgwrvJUXCz0/GU1B8pAAvQxNE
P4Htecc biJVMWwXvdMX0h5i89vqbFCMP4QM1s+3ywPgym2hFEwbid3tALBSFK+RbLE4E9Hp EgjAALAcKxHad3A2m
670eYfcgnDmCXRwVwvvo2ifv922ebPynXApVfSr/5Vh881A bx3Rvp0704gqu52/clpWcTs/1PPRCv4o76Pu2ZmvA9
OPYLfykqGxvYmJHzDNw6Yu Yj0uFgJ3RFrNgQo8p0Quebg/BLxcoIfhG69Rjs3sLPr4/m3wOnyqi+Rn1TGNAgMB AA
GjQjBAMA8GA1UdEwEB/wQFMAMBAF8wDgYDVR0PAQH/BAQDAgGGMB0GA1UdDgQW BBT0w0q5mVXyuNtgv61+vVa11za
n1iANB8kqhkiG9w0BAQsFAAOCADFvVvVfOPT QW5n16d1Fe88h17v0n3GeDeda7aikmku0Gvhf0TUIiawXMTeKvSHM
```

Certificate Verification

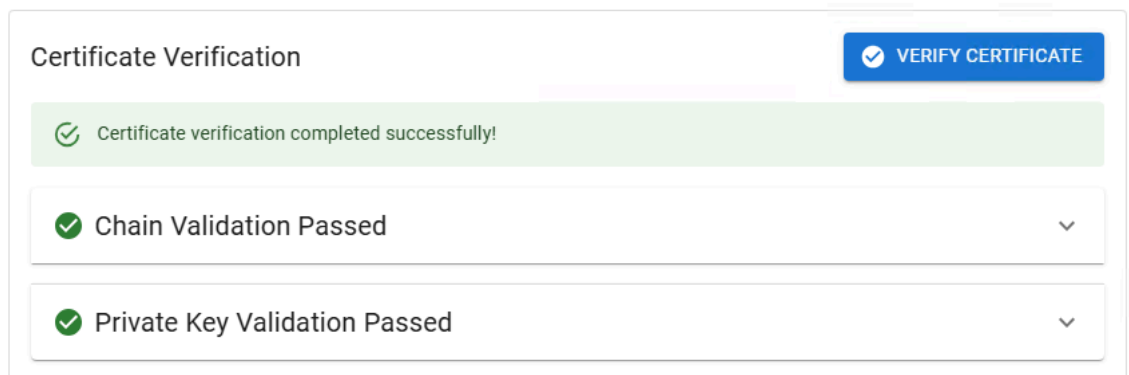
 VERIFY CERTIFICATE

Verify Certificate

The **Verify Certificate** button tasks TLS Manager with verifying the certificate.

This checks the PEM encoded data to ensure it represents a correctly formatted certificate and also, in the case of Local Key Pairs it checks that the provided key matches the provided certificate.

Verification Example:



The screenshot shows a 'Certificate Verification' interface. At the top right is a blue button with a white checkmark icon and the text 'VERIFY CERTIFICATE'. Below the button is a green success message: 'Certificate verification completed successfully!'. Underneath are two expandable sections, each with a green checkmark icon and a dropdown arrow. The first section is labeled 'Chain Validation Passed' and the second is 'Private Key Validation Passed'.

5. TLS Manager: Importing

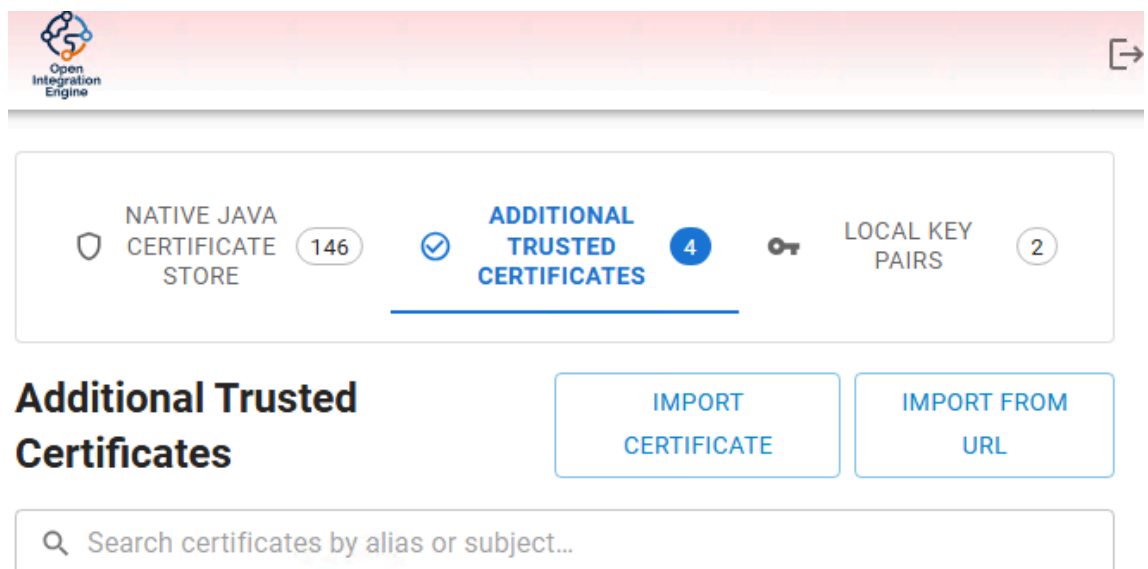
Additional Trusted Certificates

This category is for certificates that can be used by the following connector types as **Trusted Server Certificates**:

- HTTP Sender
- Web Service Sender
- TCP Sender (Client Mode)
- TCP Listener (Client Mode)

They can also be used by these connector types as **Trusted Client Certificates**:

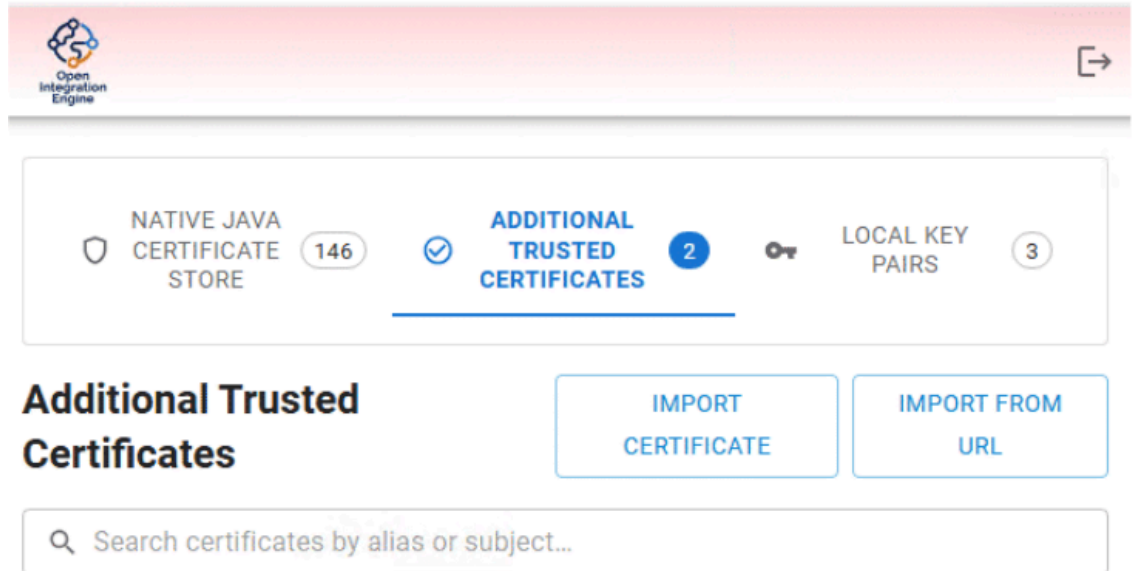
- HTTP Listener (mTLS)
- Web Service Listener (mTLS)
- TCP Listener (Server Mode) (mTLS)
- TCP Sender (Server Mode) (mTLS)



The screenshot shows the 'Open Integration Engine' interface. At the top left is the logo, and at the top right is a navigation icon. Below the header is a navigation bar with three items: 'NATIVE JAVA CERTIFICATE STORE' with a shield icon and a count of 146; 'ADDITIONAL TRUSTED CERTIFICATES' with a checkmark icon, a blue circle containing the number 4, and a blue underline; and 'LOCAL KEY PAIRS' with a key icon and a count of 2. Below the navigation bar, the 'Additional Trusted Certificates' section is active, featuring two buttons: 'IMPORT CERTIFICATE' and 'IMPORT FROM URL'. At the bottom of this section is a search bar with the placeholder text 'Search certificates by alias or subject...'.

Import Method 1: Choose Certificate File

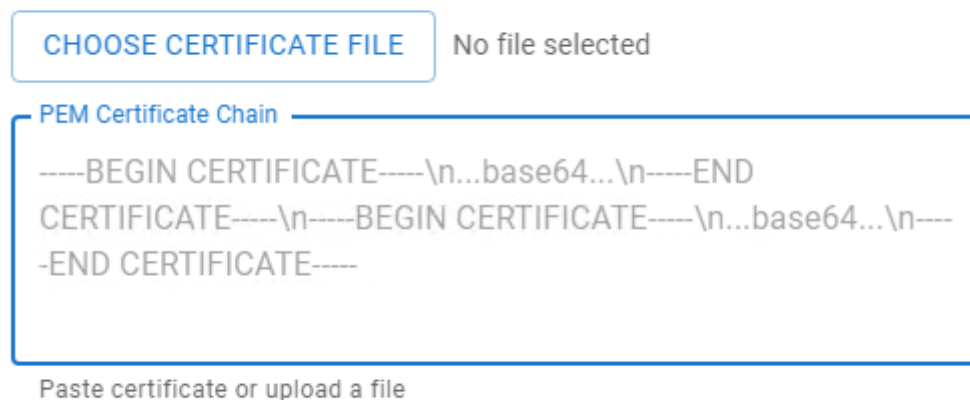
To import a certificate from file click the Import Certificate button:



The screenshot shows the 'Open Integration Engine' interface. At the top left is the logo, and at the top right is a share icon. Below this is a navigation bar with three options: 'NATIVE JAVA CERTIFICATE STORE' (146), 'ADDITIONAL TRUSTED CERTIFICATES' (2), and 'LOCAL KEY PAIRS' (3). The 'ADDITIONAL TRUSTED CERTIFICATES' option is selected and underlined. Below the navigation bar, the title 'Additional Trusted Certificates' is displayed. To the right of the title are two buttons: 'IMPORT CERTIFICATE' and 'IMPORT FROM URL'. Below these buttons is a search bar with the placeholder text 'Search certificates by alias or subject...'.

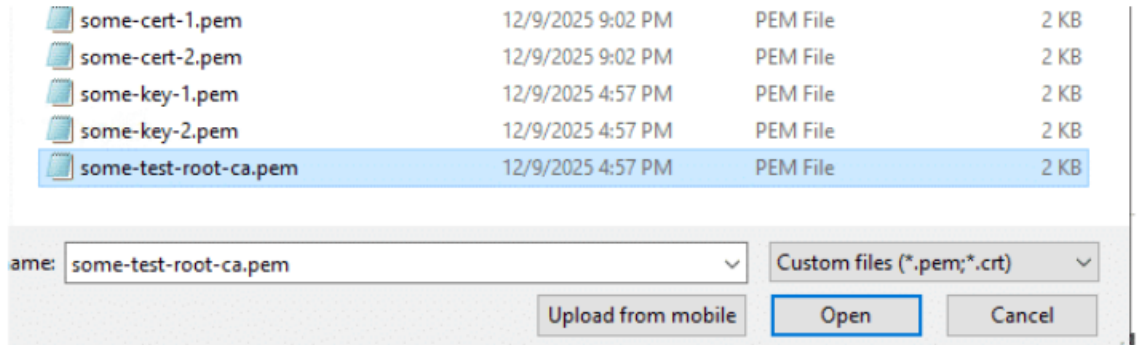
Click Choose Certificate File:

Import Certificate Chain



The screenshot shows a dialog box for importing a certificate chain. At the top, there is a button labeled 'CHOOSE CERTIFICATE FILE' and the text 'No file selected'. Below this is a text area with the title 'PEM Certificate Chain' and the following text: '-----BEGIN CERTIFICATE-----\n...base64...\n-----END CERTIFICATE-----\n-----BEGIN CERTIFICATE-----\n...base64...\n-----END CERTIFICATE-----'. At the bottom of the text area, there is a label 'Paste certificate or upload a file'.

Select your certificate. The certificate file name on disk must end in .pem or .crt:



The certificate will be decoded and the details will be displayed:

Import Certificate Chain

Select a certificate to import:

Some Test Root CA

Alias *
Some Test Root CA

PEM
-----BEGIN CERTIFICATE-----
MIIFWjCCA0KgAwIBAgIU
PJ0xGac8V+Jws+uso9jJ2Y9uYJ
QwDQYJKoZIhvcNAQEL
BQAwMzEVMBMGA1UECgw
MU29tZSB1Z3N0IENBMRow

PEM certificate.

Certificate Details

Subject
/O=Some Test CA/CN=Some Test Root CA

Issuer
/O=Some Test CA/CN=Some Test Root CA


Type
Root CA

Serial Number
3c9d3119a73c57e270b3ebaca3d8c9d98f6e6094

Validity Period
From: 2025-12-09
To: 2035-12-07

SHA-1 Fingerprint
B8BC7FCA1ACD890382786F0F6DA3A2059B96A388


Certificate Verification

 Certificate verification completed successfully!

CANCEL

IMPORT CERTIFICATE

The certificate after successful import:

 **some test root ca** Valid
Root CA

Subject:
/O=Some Test CA/CN=Some Test Root CA

Issuer:
/O=Some Test CA/CN=Some Test Root CA

Valid From: Valid To:
2025-12-09 2035-12-07

Fingerprint (SHA-1):
B8BC7FCA1ACD890382786F0F6DA3A2059B96A38B

[View Details](#) [Edit Alias](#) [Remove](#)

Import Method 2: Paste Certificate

To paste a certificate, first click the Import Certificate button:

Additional Trusted Certificates

Then copy your certificate text (including the BEGIN CERTIFICATE and END CERTIFICATE lines) and paste into the box:

```

valid.crl.caddy.crt.pem
1 -----BEGIN CERTIFICATE-----
2 MIIE2DCCAsCgAwIBAgICEAAwDQYJKoZIhvcNAQELBQAwNjELMAkGA1UEBhMCRUUx
3 EDAOBgNVBAoMB1Rlc3QgQ0ExFTATBgNVBAMMDFRlc3QgUm9vdCBDQTAeFw0yNTA5
4 MjMxMjE5NTNaFw0yNzEyMjcxMjE5NTNaMDQxEjAQBgNVBAMMCWxvY2FsaG9zdDER
5 MA8GA1UECgwIVGVzdCBPcmcxZzAJBgNVBAYTAKVMIIBIjANBgkqhkiG9w0BAQEF
6 AAOCAQ8AMIIBCgKCAQEAsnwMHGM3uAiYTZnK+J93ZrkmX2DIO24Tjt9I5qZM+yL6
7 H7/j94Rb09BE1Tuy/k8bhnYLXjoPt50bbtJX4/g6Pi7iyBjn8CBaGm0x8LYzpzX
8 FKyLKZO+Zc1pQ0AqoGni/oHHV3n82eXNoPHchg3CTbBjvkD3CK4om/pQDDfKS/8z
9 cmxw4/dilVdlp2M7vP6iMLc6ad+ha4D5T9KMJ4iJ+f/k5h24IVAJpz1GwCMOM/oL
10 Vjhr6MYirQnxghGQNWxqtHDnicz5nkiFdTsRlZ2iRA6Gj/erEG6LraVsLXdg/Hyz
11 AltCoDy1xKk7q+AqoLLt/J2L6N6A1000vHTJ0G6kUQIDAQABo4HxMIHuMAwGA1Ud
12 EwEB/wQCMAAwDgYDVR0PAQH/BAQDAgWgMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggr
13 BgEFBQcDAjAdBgNVHQ4EFgQUv6moDtbyyrdAEG0UKw3gaal3HtYwHwYDVR0jBBgw
14 FoAUHF00S3+Y6w8x+EQkzMoIzkzrfJMwLwYDVR0fBCgwJjAkoCKgIIYeaHR0cDov
15 L2V4YW1wbGUudGVzdC9jcmwvY2EuY3JsMD4GA1UdEQQ3MDWCCWxvY2FsaG9zdIIP
16 dmFsaWQuY3JsLmNhZGR5ghFyZXZva2VklmNybc5jYWRkeYcEfwAAATANBgkqhkiG
17 9w0BAQsFAA0CAgEAdtc3Dr2q9RtY4SSn45rFUw4ZzXq+q12TBqEzypWax7VvNWFE
18 em/1tsrXmwSzc3z4t4gc9e3brYFnnGXnoHkyZvyf7Xn24Yu99oHck++KL5LmHu2
19 KJRwFrtqK/QNuFrZaVSiCo0wz5M1Da5WvyVgm+8QbkhSRmWFjxmzjRHVjs4e9s+
20 fQEoNUBpDwq2UikoLFQgXJ9xJ4BYn9Z0/NDKQ3WMyFcQjIFxjp/iUnw3VJ/GTRC1
21 GmN6eAfGCfjbXipVNVEzpTKAZ3gySAtpZR4Uw6WMSXwZTpfStFIkjZm1V4mWVsnQ
22 l3oj9eqjirRHTeCc+RiawWga7H1pwPgpe0rKRiCVqrPp/9c+ydM+3nCyaJiyRtJx
23 P+i0jAQTYXEln6NmfvYGe0UBlib+68hZTTzk7xorVv1rkeHsD+qsnJ00MroOnrG
24 PwAL9TD0tqKMBXrKPEo4J50GiEtmYrufdqzEPOzwWj1LSfYQszhu5x8h6UI30V8Q
25 ejpANwrnmV5lW2eOqR3lvnv2emPehQq9bPXD2qJdU2oWpB6EE8WLfzoHgjdHIEw
26 R9FceJYYbYEaIP1StrIzalZtSB7ofrBFUL6Wq+GJR7VXBpKhFbU0W7bm64GlszIo
27 oUx12UD5ED6H/QEa/Bvy/xyKjg62teVinS4ZzCx7kBFyLSZELEukK8t/XgU=
28 -----END CERTIFICATE-----

```

The certificate will be decoded and the details will be displayed:

Import Certificate Chain

CHANGE CERTIFICATE FILE
some-cert-2.pem

PEM Certificate Chain

```

-----BEGIN CERTIFICATE-----
MIIE2TCCAsGgAwIBAgICEAEwDQYJKoZIhvcNAQELBQAwMzEVMBMGA1UECgwMU29t
ZSBUZXN0IENBMRowGAYDVQQDDDBFTb21lIFRlc3QgUm9vdCBDQTAeFw0yNTEyMDkx
NDU3MzFaFw0yODAzMTMxNDU3MzFaMCwxEjAQBgNVBAMMCWxvY2FsaG9zdDEWMBQG
A1UECgwNU29tZSBUZXN0IE9yZzCCASlwdQYJKoZIhvcNAQEBBQADggEPADCCAQoC
ggEBALwrFAwI1EW2rGOyDFqmdVZT3D/RbaVSVgHjk2r9AtZGErHyRatbZ/fg8iG
UsPO/sejhyCgpPpxBZLq5YP6I9EcvTTqIIOfpSklu60HBomaXHIQ2TxHNNBoDX3d

```

Paste certificate or upload a file

✓
Found 1 certificate in the chain

Select a certificate to import:

●
localhost/O=Some Test Org

The imported certificate:

🛡️

some cert 2

End-entity

Valid

Subject:

/CN=localhost/O=Some Test Org

Issuer:

/O=Some Test CA/CN=Some Test Root CA

Valid From: Valid To:

2025-12-09 2028-03-13

Fingerprint (SHA-1):

0C7ACBA640FC037C9B4B7989D51E2582245ABEF4

🛡️
View Details

✎
Edit Alias

🗑️
Remove

Import Method 3: Import From URL

To import a certificate from a URL click the **Import From URL** button:

Additional Trusted Certificates

[IMPORT CERTIFICATE](#)[IMPORT FROM URL](#)

Paste in a HTTPS URL, including the port if required:

Import Certificate from URL

URL

[FETCH
CERTIFICATES](#)

Enter a valid HTTPS URL to fetch certificates

Click **Fetch Certificates**.

The URL may offer more than one certificate. Select one:

Import Certificate from URL

URL FETCH CERTIFICATES

Enter a valid HTTPS URL to fetch certificates

Select a certificate to import:

www.nhs.uk

DigiCert Global G2 TLS RSA SHA256 2020 CA1

Alias*

PEM

PEM certificate.

i Certificate Details

Subject
/C=US/O=DigiCert Inc/CN=DigiCert Global G2 TLS RSA SHA256 2020 CA1

Issuer
/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=DigiCert Global Root G2


Type
Intermediate


Serial Number
0cf5bd062b5602f47ab8502c23ccf066


Validity Period
From: 2021-03-30
To: 2031-03-29

Click Import Certificate:

SHA-1 Fingerprint
1B511ABEAD59C6CE207077C0BF0E0043B1382612


 **Certificate Verification**

 Certificate verification completed successfully!

 **Chain Validation Passed** ▼

CANCEL
IMPORT CERTIFICATE

The imported certificate:



digicert global g2 tls rsa sha256 2020 ca1
Intermediate


Valid


Subject:
/C=US/O=DigiCert Inc/CN=DigiCert Global G2 TLS RSA SHA256 2020 CA1


Issuer:
/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=DigiCert Global Root G2

Valid From: Valid To:
2021-03-30 2031-03-29

Fingerprint (SHA-1):
1B511ABEAD59C6CE207077C0BF0E0043B1382612

 View Details

 Edit Alias

 Remove

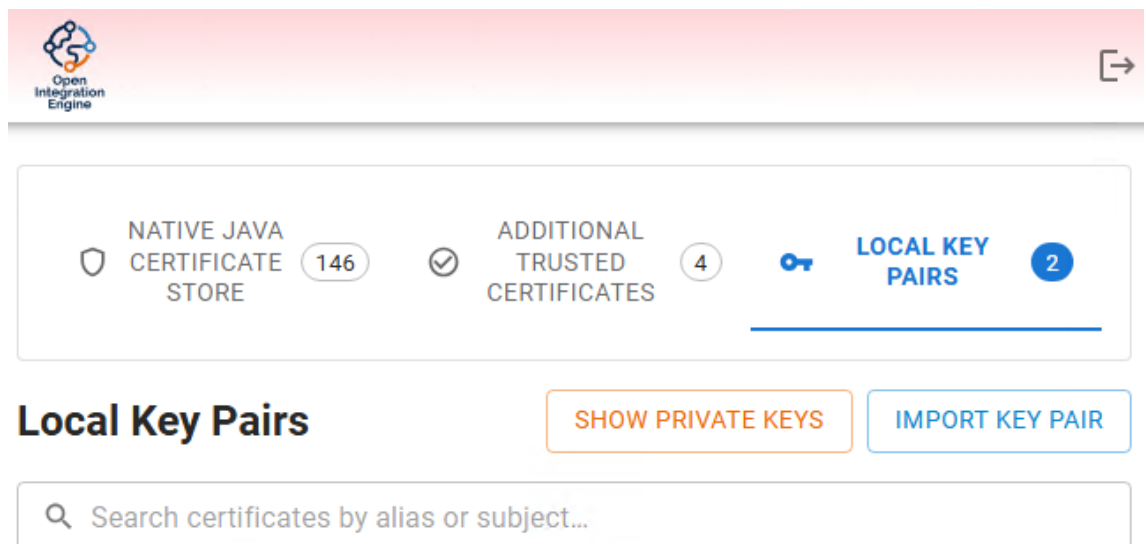
Local Key Pairs

This category is for certificates that can be used by the following connector types as **Client Certificates**:

- HTTP Sender (mTLS)
- Web Service Sender (mTLS)
- TCP Sender (Client Mode) (mTLS)
- TCP Listener (Client Mode) (mTLS)

They can also be used by these connector types as **Server Certificates**:

- HTTP Listener
- Web Service Listener
- TCP Sender (Server Mode)
- TCP Listener (Server Mode)



The screenshot shows the 'Open Integration Engine' interface. At the top left is the logo, and at the top right is a share icon. Below the header is a navigation bar with three items: 'NATIVE JAVA CERTIFICATE STORE' with a shield icon and a count of 146, 'ADDITIONAL TRUSTED CERTIFICATES' with a checkmark icon and a count of 4, and 'LOCAL KEY PAIRS' with a key icon and a count of 2. The 'LOCAL KEY PAIRS' item is currently selected and underlined. Below the navigation bar, the 'Local Key Pairs' section is active, featuring a search bar with the placeholder text 'Search certificates by alias or subject...'. To the right of the search bar are two buttons: 'SHOW PRIVATE KEYS' and 'IMPORT KEY PAIR'.

Import Key Pair

To import a local certificate click **Import Key Pair**.

The **Alias** will be auto-populated but can be edited.

Click **Choose Certificate File** and select the public key.

Click **Choose Private Key File** and select the private key.

Only .pem or .crt files are accepted.

Import Key Pair (PEM)

Alias *

CHOOSE CERTIFICATE FILE

No file selected

PEM (paste contents including BEGIN/END)

Paste PEM certificate. Uploading a .pem or .crt file fills this field.

CHOOSE PRIVATE KEY FILE

No private key file selected

Private Key (paste contents including BEGIN/END) *

Paste private key. Uploading a .pem or .key file fills this field.

In this example we have selected a key pair and set an alias:

Import Certificate (PEM)

Alias *

[CHANGE CERTIFICATE FILE](#) other-cert-1.pem

PEM (paste contents including BEGIN/END)

Paste PEM certificate. Uploading a .pem or .crt file fills this field.

[CHANGE PRIVATE KEY FILE](#) other-key-1.pem

Private Key (paste contents including BEGIN/END) *

Paste private key. Uploading a .pem or .key file fills this field.

i Certificate Details

Subject
/CN=localhost/O=Other Test Org

Issuer
/O=Other Test CA/CN=Other Test Root CA

Type
End-entity

Serial Number
1000

Validity Period
From: 2025-12-09
To: 2028-03-13

SHA-1 Fingerprint
3401703F64536618CF2E637E44835AE9D6329378

Subject Alternative Names [Show More](#) ▼

DNS Names:


- localhost
- valid.crl.caddy
- revoked.crl.caddy
- mtls.caddy

IP Addresses:

- 127.0.0.1

Click **Import Certificate**.

The imported certificate:




 **other cert 1**
End-entity Valid

Subject:
/CN=localhost/O=Other Test Org

Issuer:
/O=Other Test CA/CN=Other Test Root CA

Valid From: Valid To:
2025-12-09 2028-03-13

Fingerprint (SHA-1):
3401703F64536618CF2E637E44835AE9D6329378


 View Details Edit Alias Remove

Show Private Keys

This button expands the panel display for all Local Key Pairs to show Private Keys:

[SHOW PRIVATE KEYS](#)

For example:



other cert 1

End-entity

Valid

Subject:
/CN=localhost/O=Other Test Org


Issuer:
/O=Other Test CA/CN=Other Test Root CA


Valid From: Valid To:
2025-12-09 2028-03-13


Fingerprint (SHA-1):
3401703F64536618CF2E637E44835AE9D6329378

Private Key (Base64):

```
-----BEGIN PRIVATE KEY----- MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBCwggSjAgEAAoIBAQC/
qTdI1gyrncPf 1DsL3Fdi40AsxniP6JsdI+Trh4LTAAofi0iCfok+dvKa4F0aWkbGREQUqx179v7a IR
3P89Q3CdEY6bxGqNxxCdnP333ai6jLS0/y91IySxIKyTW1rv6tf4qSCuBhL9uW 4EvZfP+SWg9TgSdqM
He2+vHKJmbyxacRhbQZmiXggF6TXC9uX1PPFGnvCWMDNy6K OTwd6Kude+XOjpvfUpYzb9uUvFRtnc8B
Z1zt/NQfED2QjXZ1gA6xDatHQD/TczI/ m8v0FwisglqzhGajpv75XY17EQk3bEx94vrw4e583+UWbqr
HkUJHPvLVbDpM/2o1 yYgJ7kpNAgMBAACggEALChFma7KdAa48HqR8RFeu8wCMia287b8d9oDfcVenZ
```

 [View Details](#)

 [Edit Alias](#)

 [Remove](#)

6. TLS Manager: Common Tasks

Certificate Rename

A certificate alias is automatically populated when the certificate is imported.

It can be edited by clicking the **Edit Alias** button, editing it and clicking **Update Alias** to save.

If the certificate is used by a channel under its alias, a warning will be displayed:

Edit Certificate Alias

 This certificate is currently in use by the following channels (1):

`nmh_webservice_listener_TLS`

Changing the alias may affect channel configurations. Please ensure channels are updated accordingly.

New Alias *

other cert 1

Enter a unique alias for this certificate

Certificate Information

Current Alias	Store	Subject
other cert 1	Private	/CN=localhost/O=Other Test Org

Issuer	/O=Other Test CA/CN=Other Test Root CA
--------	--


CANCEL

UPDATE ALIAS

To update click Update Alias.

Expiry Warning

When a certificate is within 30 days of its expiry date you will see a warning indicating how many days are remaining.

 **expiring.certificate** Expires in 14 days
End-entity

Subject:
/CN=expiring.certificate

Issuer:
/C=EE/O=Test CA/CN=Test Root CA

Valid From: 2025-02-06 Valid To: 2026-02-05

Fingerprint (SHA-1):
49AB537A892A72EA9E69CF0980EB9BC4BD1FB5FD

[View Details](#) [Edit Alias](#) [Remove](#)

Certificate Replacement

When selecting a certificate for import the system will check if an existing certificate has the same Alias.

If it finds match a warning will be displayed:

Alias * _____
 localhost/O=Other Test Org

This alias is already in use in this store

If you continue with certificate import you will be advised that you will be replacing the certificate with the same Alias.

The name of any channels using the certificate will be displayed.

Any deployed channels using the certificate will need to be re-deployed to pick up the replacement certificate.

If you want to proceed click **Replace Certificate**:

Replace Existing Certificate

A certificate with the alias "other cert 2" already exists in the private store. This will replace the existing certificate. Are you sure you want to continue?

Certificate that will be replaced:

Alias	Subject
other cert 2	/CN=localhost/O=Other Test Org
Issuer	
	/O=Other Test CA/CN=Other Test Root CA

⚠ This certificate is currently in use by the following channels (1):

1_http_listener_TLS

Replacing this certificate may affect channel configurations. Please ensure channels are updated accordingly.

CANCEL

REPLACE CERTIFICATE

Certificate Removal

To remove a certificate click **Remove**.

If the certificate to be removed is used in a channel the channel details will be displayed.


You will be asked to confirm you want to **Remove Certificate**:

Remove Certificate

Certificate Information


Alias	Store	Subject
other cert 2	Private	/CN=localhost/O=Other Test Org

Issuer
/O=Other Test CA/CN=Other Test Root CA

 This certificate is currently in use by the following channels (1):

[1_http_listener_TLS](#)

Removing this certificate may affect channel configurations. Please ensure channels are updated accordingly.



 This action cannot be undone. The certificate will be permanently removed from the private store.

Are you sure you want to remove this certificate?

[CANCEL](#)

[REMOVE CERTIFICATE](#)

Once removed the successful removal notification will appear:

 Certificate "google trust services" has been removed successfully 

7. Sender TLS Settings

These settings are available to the following connector types:

- HTTP Sender
- Web Service Sender
- TCP Sender

Use TLS Manager: *Yes/No*

Use TLS Manager: Yes No

Enabling the TLS Manager permits a greater degree of control over the TLS connections.

When enabled, the connector will use the settings below for TLS connections, else the JVM settings will be used.

Subject DN Validation Mode: *None/Partial/Exact*

Subject DN Validation Mode:

CRL Mode:

OCSP Mode:

A way of filtering using Distinguished Name attributes.

Partial: Distinguished Name must contain the specified RDN:

Example DN: CN=kaur, OU=dev, C=estonia

Successful matches:

RDN: C=estonia; RDN: OU=dev CN=kaur

Unsuccessful match:

RDN: C=estonia OU=dev CN=polaris

Exact: Distinguished Name must match the specified RDN:

Example DN: CN=kaur, OU=dev, C=estonia

Successful match:

RDN: CN=kaur, OU=dev, C=estonia

CRL Mode: *Disabled/Soft Fail/Hard Fail*

CRL Mode:	Disabled ▾
OCSP Mode:	Disabled Soft Fail
Enabled Protocols:	Hard Fail ed

To have the endpoint certificate automatically checked against a Certificate Revocation List (CRL) service.

The service used is defined within each certificate. It will typically be the Issuing Authority.

CRL responders are contacted using the protocol and port implied (or explicitly stated) in the URLs embedded in the certificate.

- **Soft Fail** will allow the connection to proceed in the event that the CRL service does not respond.
- **Hard Fail** will only allow the connection if the CRL service responds and the certificate is confirmed as not revoked.

OCSP Mode: *Disabled/Soft Fail/Hard Fail*

OCSP Mode:	Disabled ▾
Enabled Protocols:	Disabled ed Soft Fail
Enabled Ciphers:	Hard Fail ed

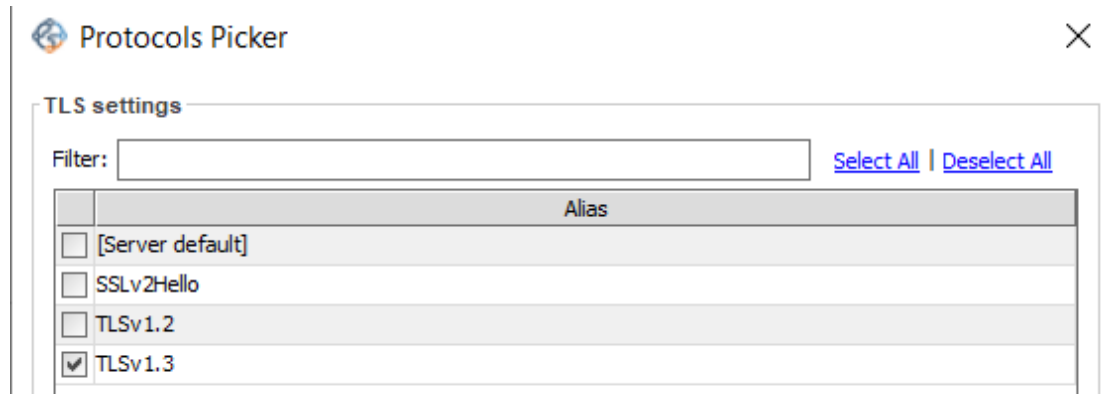
To have the endpoint certificate automatically checked against an Online Certificate Status Protocol service (OSCP).

The service used is defined within each certificate. This is typically the Certificate Authority (CA) that issued it.

OCSP responders are contacted using the protocol and port implied (or explicitly stated) in the URLs embedded in the certificate.

- **Soft Fail** will allow the connection to proceed in the event that the CRL service does not respond.
- **Hard Fail** will only allow the connection if the CRL service responds and the certificate is confirmed as not revoked.

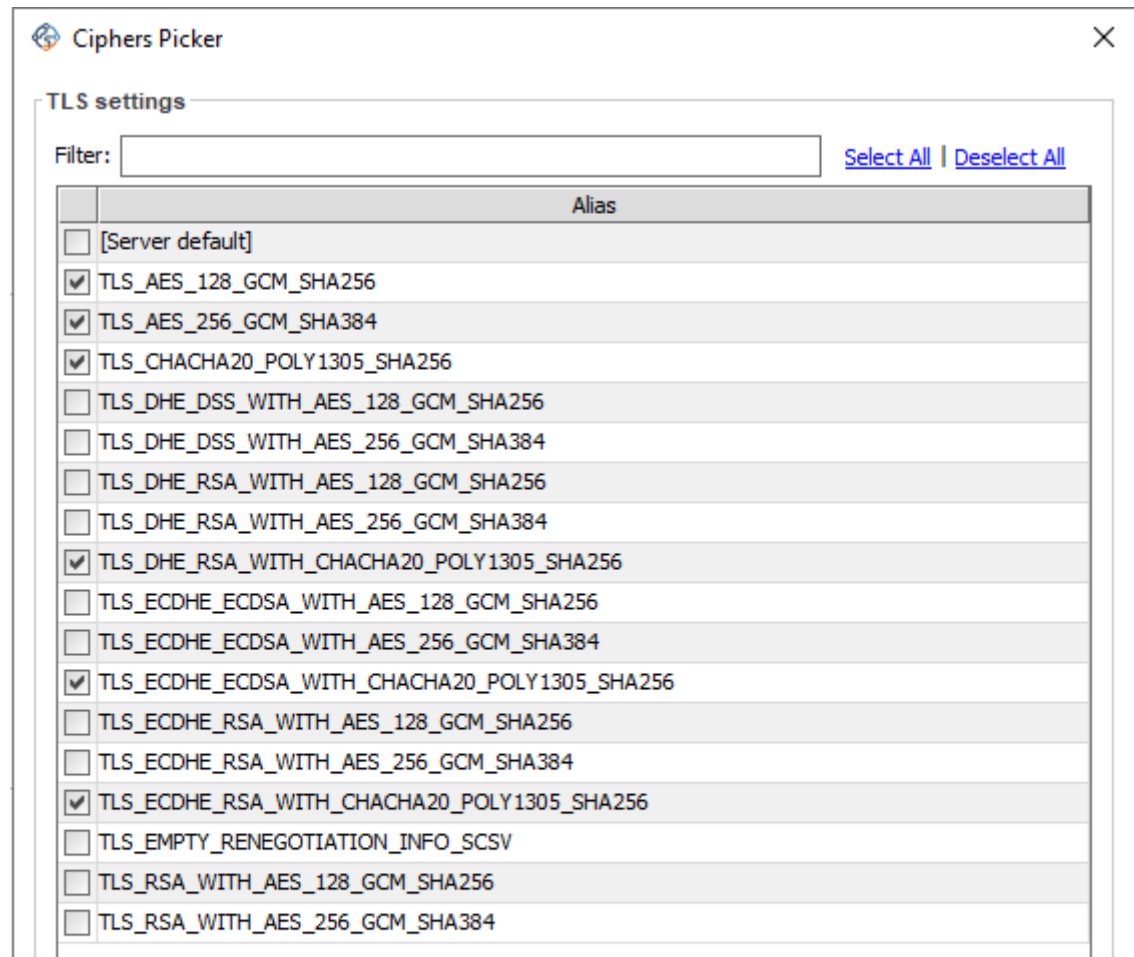
Enabled Protocols:



Select the acceptable protocol or protocols.

The list is defined in mirth.properties as https.server.protocols


Enabled Ciphers:

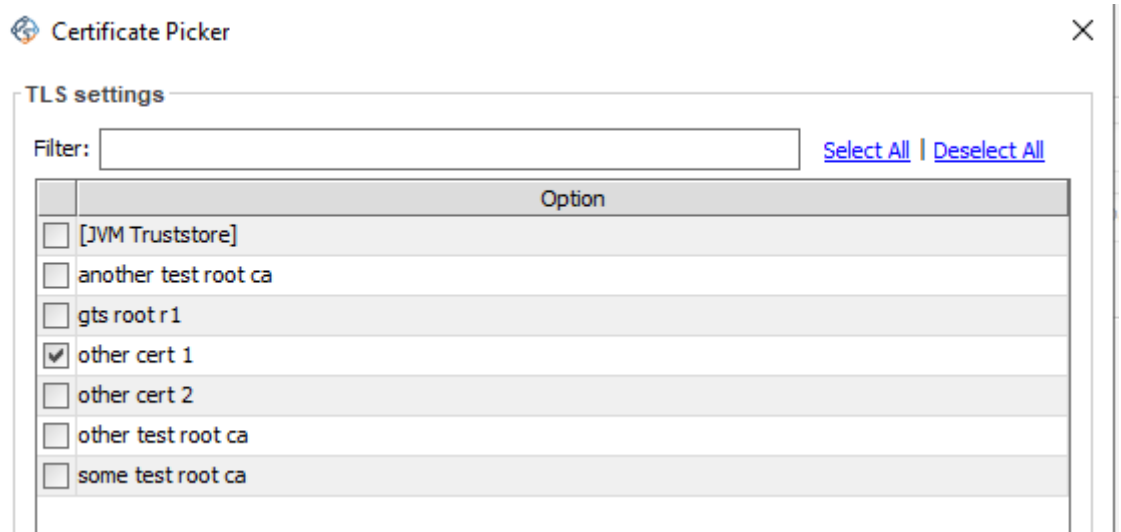


Select the ciphers associated with the selected protocol(s).

The list is defined in mirth.properties as https.ciphersuites

Trusted Server Certificates:

Trusted Server Certificates:  Trusting 1 certificate



Enabled for TCP Sender in Client Mode only.

The trusted server. Select from **Additional Trusted Certificates**.

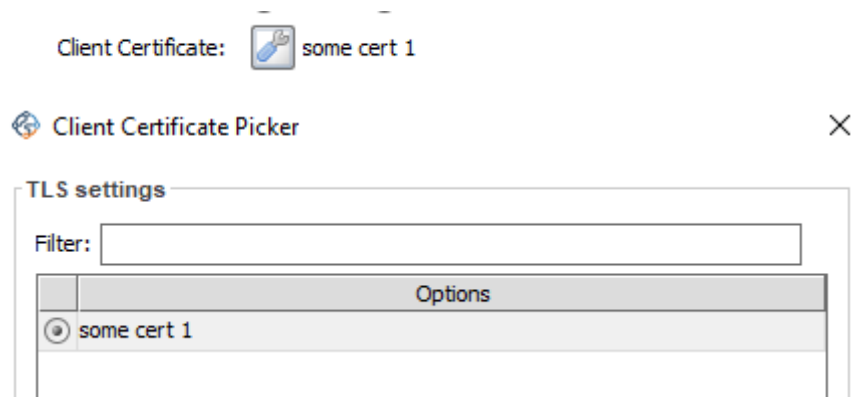
Hostname verification: *Enabled/Disabled*

Hostname verification: Enabled Disabled

Enabled for TCP Sender in Client Mode only.

Performs hostname verification by checking that the certificate's Common Name (CN) or Subject Alternative Names (SAN) match the requested endpoint.

Client Certificate:




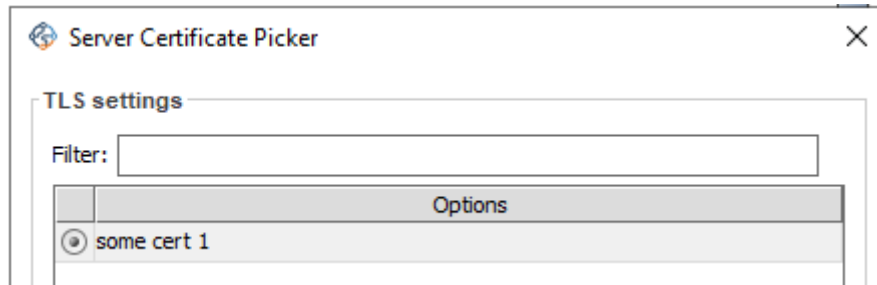
Enabled for TCP Sender in Client Mode only.

Your client certificate to present to the endpoint if requested or required (mTLS). Select from Local Key Pairs.

Exclusive to TCP Sender in Server Mode

Server Certificate:

Server Certificate:  some cert 1




Your server certificate. Select from Local Key Pairs.

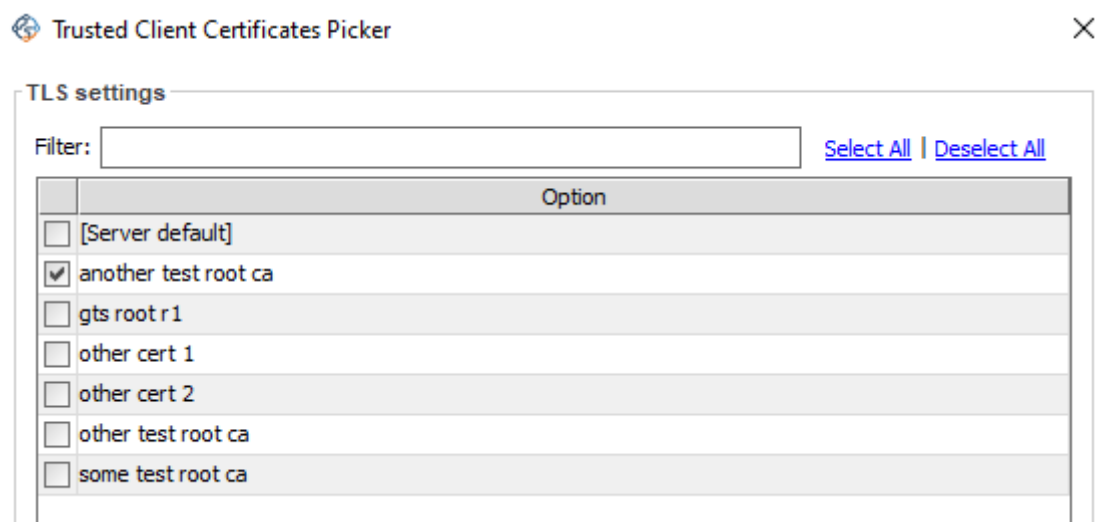
Client Authentication Mode:

Client Authentication Mode None Requested Required

If you want to request or require a client certificate (mTLS).

Trust Client Certificate:

Trusted Client Certificates:  Trusting 1 certificate



The certificate of the entity acting as the client (mTLS). Select from Additional Trusted Certificates.

8. Listener TLS Settings

These settings are available to the following connector types:

- HTTP Listener
- Web Service Listener
- TCP Listener

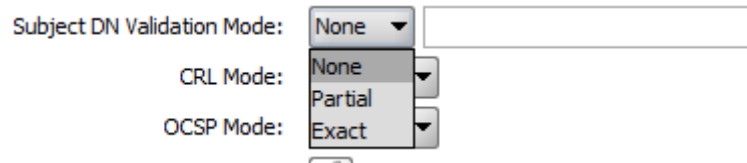
Use TLS Manager: *Yes/No*

Use TLS Manager: Yes No

Enabling the TLS Manager permits a greater degree of control over the TLS connections.

When enabled, the connector will use the settings below for TLS connections, else the JVM settings will be used.

Subject DN Validation Mode: *None/Partial/Exact*



The screenshot shows a configuration interface for Subject DN Validation Mode. It includes three dropdown menus: 'Subject DN Validation Mode' (set to 'None'), 'CRL Mode' (set to 'None'), and 'OCSP Mode' (set to 'Exact'). There is also an empty text input field next to the first dropdown.

A way of filtering using Relative Distinguished Names (RDN).

None: No filter

Partial: Distinguished Name must contain the specified RDN::

Example DN: CN=kaur, OU=dev, C=estonia

Successful matches:

RDN: C=estonia; RDN: OU=dev CN=kaur

Unsuccessful match:

RDN: C=estonia OU=dev CN=polaris

Exact: Distinguished Name must match the specified RDN:

Example DN: CN=kaur, OU=dev, C=estonia

Successful match:

RDN: CN=kaur, OU=dev, C=estonia

CRL Mode: *Disabled/Soft Fail/Hard Fail*

CRL Mode:	Disabled ▾
OCSP Mode:	Disabled
Enabled Protocols:	Soft Fail
	Hard Fail

To have the endpoint certificate automatically checked against a Certificate Revocation List (CRL) service.

The service used is defined within each certificate. It will typically be the Issuing Authority.

CRL responders are contacted using the protocol and port implied (or explicitly stated) in the URLs embedded in the certificate.

- **Soft Fail** will allow the connection to proceed in the event that the CRL service does not respond.
- **Hard Fail** will only allow the connection if the CRL service responds and the certificate is confirmed as not revoked.

OCSP Mode: *Disabled/Soft Fail/Hard Fail*

OCSP Mode:	Disabled ▾
Enabled Protocols:	Disabled
	Soft Fail
Enabled Ciphers:	Hard Fail

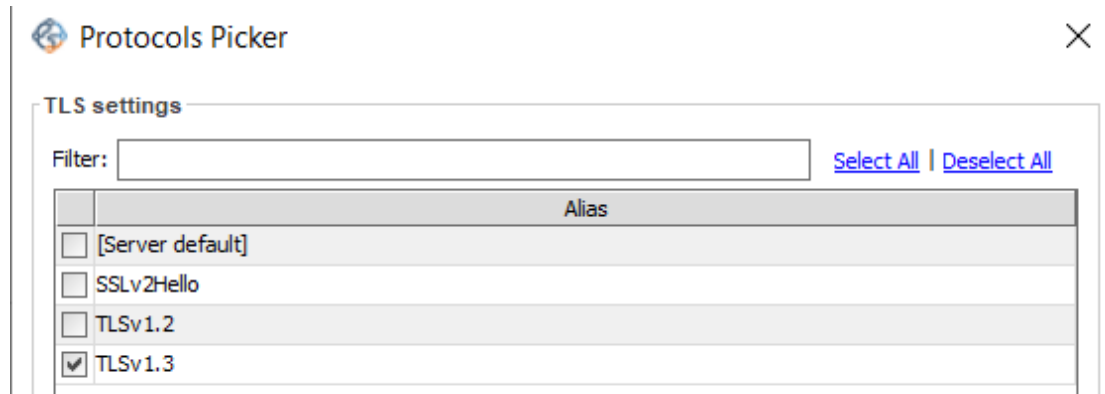
To have the endpoint certificate automatically checked against an Online Certificate Status Protocol service (OSCP).

The service used is defined within each certificate. This is typically the Certificate Authority (CA) that issued it.

OCSP responders are contacted using the protocol and port implied (or explicitly stated) in the URLs embedded in the certificate.

- **Soft Fail** will allow the connection to proceed in the event that the CRL service does not respond.
- **Hard Fail** will only allow the connection if the CRL service responds and the certificate is confirmed as not revoked.

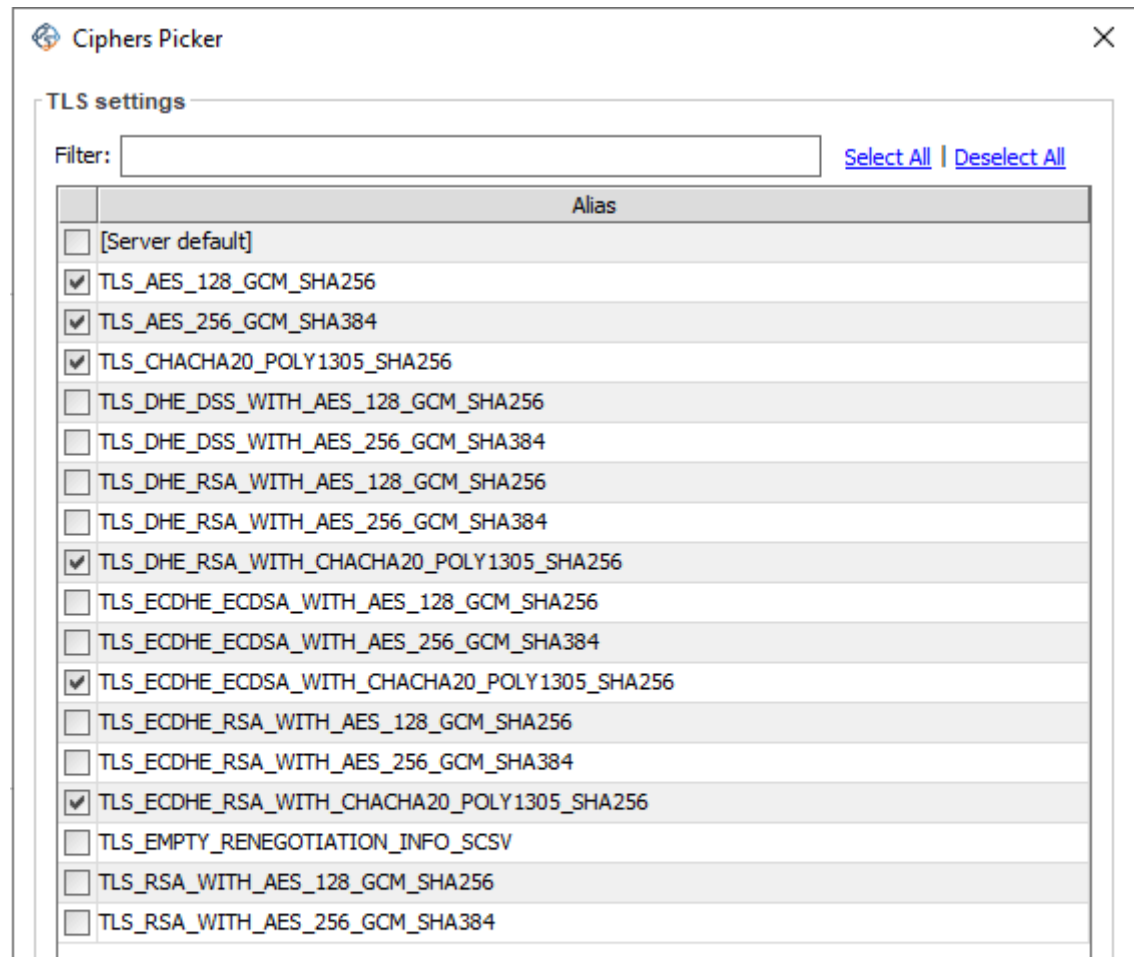
Enabled Protocols:



Select the acceptable protocol or protocols.

The list is defined in mirth.properties as https.server.protocols

Enabled Ciphers:

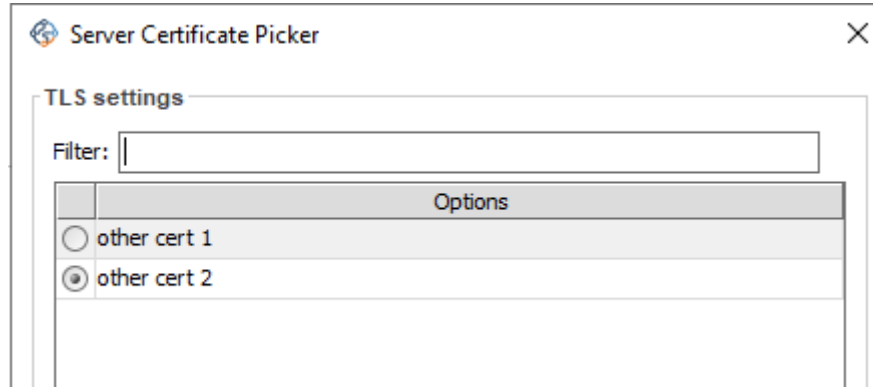


Select the ciphers associated with the selected protocol(s).

The list is defined in mirth.properties as https.ciphersuites

Server Certificate:

Server Certificate:  other cert 2



Enabled for TCP Listener in Server Mode only.

Your server certificate. Select from **Local Key Pairs**.


Client Authentication Mode: *None/Requested/Required*


Client Authentication Mode None Requested Required

Enabled for TCP Listener in Server Mode only.

If you want the client to prove their identity you can request or require that they supply a certificate (mTLS).

Trusted Client Configuration:

Trusted Client Certificates:  Trusting 1 certificate

 Trusted Client Certificates Picker ✕

TLS settings

Filter: [Select All](#) | [Deselect All](#)

	Option
<input type="checkbox"/>	[Server default]
<input type="checkbox"/>	other test root ca
<input checked="" type="checkbox"/>	some cert 1
<input type="checkbox"/>	some cert 2
<input type="checkbox"/>	some test root ca


Enabled for TCP Listener in Server Mode only.

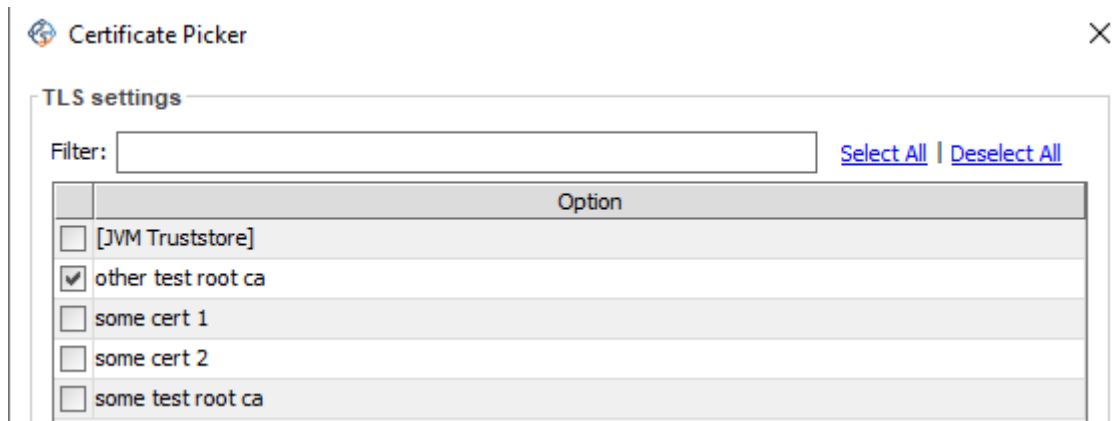
If you are requesting or requiring a Client Certificate (above), select the client certificate(s) you trust (mTLS).

Select from **Additional Trusted Certificates**.

Exclusive to TCP Listener in Client Mode

Trusted Server Certificate:

Trusted Server Certificates:  Trusting 1 certificate




The certificate associated with the server. Select from **Additional Trusted Certificates**.

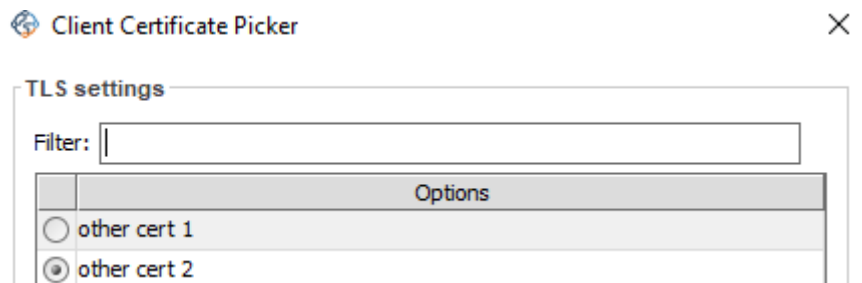
Hostname verification: *Enabled/Disabled*

Hostname verification: Enabled Disabled

Verifies that the certificate presented by the endpoint contains the correct domain information (either in Common Name (CN) and Subject Alternative Names (SAN))

Client Certificate:

Client Certificate:  other cert 2



Your client certificate to present to the endpoint if requested or required (mTLS). Select from **Local Key Pairs**.

Validation

If any TLS Settings fail validation the setting or settings that need to be addressed will be highlighted in red:

TLS Settings

Use TLS Manager: Yes No

Subject DN Validation Mode: None

CRL Mode: Soft Fail

OCSP Mode: Disabled

Enabled Protocols: 🔑 0 selected

Enabled Ciphers: 🔑 0 selected

Trusted Server Certificates: 🔑

Hostname verification: Enabled Disabled

Client Certificate: 🔑

Server Certificate: 🔑 some cert 1

Client Authentication Mode None Requested Required

Trusted Client Certificates: 🔑 None selected

9. API

URL

The TLS Plugin extends the OIE API with Swagger documented list of calls available at <https://<servername>:8443/api/#/TLS%20Manager>

Access

Access credentials are the same ones used to log in to the OIE Administrator Client:



The screenshot shows the login interface for the Open Integration Engine. It features a header with the OIE logo and the text "Open Integration Engine". Below the header, there are two input fields: "username" and "password". To the right of the "password" field is a green "Sign In" button.

API Calls

POST [/tlsmanager/_cacheWsdIFromUrl](#)

A dependency of the Get Operations button in the Web Service Sender Connector. Downloads the WSDL at the specified URL and caches the web service definition tree.

POST [/tlsmanager/_isWsdICached](#)

A dependency of the Get Operations button in the Web Service Sender Connector. Returns true if the definition tree for the WSDL is cached by the server.

GET [/tlsmanager/clientcertificates](#)

Deprecated, for removal.

POST [/tlsmanager/getDefinition](#)

Retrieves the definition service map corresponding to the specified WSDL.

GET [/tlsmanager/importedcertificates](#)

Deprecated, for removal.

GET [/tlsmanager/keystore](#)

Deprecated, for removal.

GET [/tlsmanager/localCertificates](#)

Retrieve the list of Local Key Pairs imported into the key store.

PUT [/tlsmanager/localCertificates](#)

Overwrite the local certificates within the in use keystore

GET [/tlsmanager/remoteCertificates](#)

Retrieve the chain of certificates presented at a remote URL.

GET [/tlsmanager/systemCertificates](#)

Retrieve the list of trusted certificates from the Java system trust store.

POST [/tlsmanager/testHttpsConnection](#)

Tests whether a connection can be successfully established to the destination endpoint.

POST [/tlsmanager/testTcpConnection](#)

Tests whether a connection can be successfully established to the destination endpoint.

POST /tlsmanager/testWsConnection

Tests whether a connection can be successfully established to the destination endpoint.

GET /tlsmanager/trustedCertificates

Retrieve the list of trusted certificates imported into the Additional trust store.

PUT /tlsmanager/trustedCertificates

Overwrite the trusted certificates within the in use truststore

POST /tlsmanager/truststore

Overwrite the in use truststore

10. Known Issues

A. Deselection of mTLS in Client Mode of TCP Listener and TCP Sender.

When configuring TLS Settings, if a Client Certificate is selected but you subsequently decide you want to deselect the Client Certificate and not use mTLS, it is currently not an option.

Workaround:

Change the connector to a different type and then change it back to the connector type you need to use. This will lose any configuration in the TLS settings but will facilitate fresh TLS config without mTLS.

B. Certificate Alias and Upper Case

Although the Certificate Alias field will permit upper case letters to be typed when importing a certificate or changing the alias, all aliases are stored in lower case once imported and all alias name comparison is case-insensitive.

For the latest Known Issues please visit the NovaMap repository at:

<https://github.com/NovaMap-Health/tls-manager-plugin>

Acknowledgements

Joint sponsors:

Diridium Technologies

NovaMap Health

The team members involved:

Alex Frîncu

Andreea Dincă

Andrei Haiducu

Ed Riordan

Kaur Palang

Paul Coyne

Paul Hristea

Paul Richardson

Appendix A

mirth.properties

https.client.protocols

Sets the TLS protocols supported by OIE for client operations.

Example: `TLSv1.3`

https.server.protocols

Sets the TLS protocols supported by OIE for server operations.

Example: `TLSv1.3`

Https.ciphersuites

Sets the TLS protocols supported by OIE for server operations.

Example: `TLS_AES_128_GCM_SHA256, TLS_AES_256_GCM_SHA384`

Server.http.enable

Enables the http for `/tls-manager` and `/api`

Example: `true`

Default: `false`.

Note: Please refer to your cyber security policy regarding appropriate configuration of TLS protocols and cipher suites.

Appendix B

Environment Variables

OIE_TLS_PLUGIN_PERSISTENCE_BACKEND

Possible values: `database`, `filesystem`

Case insensitive.

Defaults to `database`

OIE_TLS_PLUGIN_FS_TRUSTSTOREPATH

Possible values: Any path, absolute or relative to `oie jvm`.

Should support OS-specific formats through JVM (Win \ vs Unix /).

Case sensitivity depends on the OS

No default.

OIE_TLS_PLUGIN_FS_TRUSTSTOREPASS

Possible values: Any string, case sensitive

No default.

OIE_TLS_PLUGIN_FS_KEYSTOREPATH

Possible values: Any path, absolute or relative to OIE JVM.

Should support OS-specific formats through JVM (Win \ vs Unix /).

Case sensitivity depends on the OS.

No default.

OIE_TLS_PLUGIN_FS_KEYSTOREPASS

Possible values: Any string, case sensitive

No default.

OIE_TLS_PLUGIN_DISABLE_UI

Possible values: `true`, `false`

Known to work on Linux and macOS.

Alternative boolean representations may be required (OS dependant)

Defaults to `false`

Appendix C

Third Party Java Libraries

cxf-core-4.1.3

cxf-rt-wsdl-4.1.3

stax2-api-4.2.2

woodstox-core-7.1.1.jar

Appendix D

IETF TLS Protocols

RFC8446: The Transport Layer Security (TLS) Protocol Version 1.3

RFC5246: The Transport Layer Security (TLS) Protocol Version 1.2

Document History

Version	Plugin Version	Notes	Author	Date
1.1	1.0.0	Restructure	ETR	19-Feb-2026
1.0	1.0.0	Known Issues, Launchers, DN, Cert Change	ETR	09-Jan-2026
0.9	1.0.0	Revisions following review	ETR	08-Jan-2026
0.8	1.0.0	Certificate Category to Connector text	ETR	08-Jan-2026
0.7	1.0.0	TLS Settings: Listeners	ETR	08-Jan-2026
0.6	1.0.0	TLS Settings: Senders	ETR	08-Jan-2026
0.5	1.0.0	API, mirth.properties, env vars, replace cert	ETR	07-Jan-2026
0.4	1.0.0	Certificate Management update	ETR	07-Jan-2026
0.3	1.0.0	Installation chapter	ETR	06-Jan-2026
0.2	1.0.0	PR Comments	ETR	06-Jan-2026
0.1	1.0.0	Initial Document	ETR	05-Jan-2026