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.

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

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

a) Web based administration portal for certificate management.

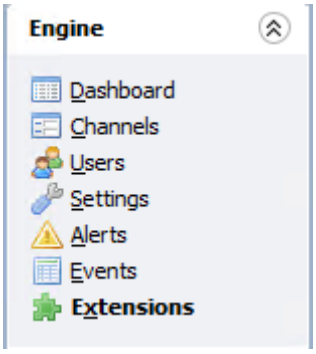
b) A REST API for certificate management, documented according to the OpenAPI specification.

c) TLS and mTLS capability in HTTP, TCP and Web Services connector types.

2. Installation

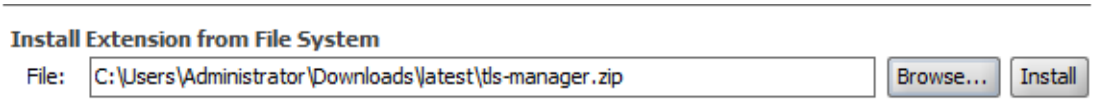
The plugin is available to download from the NovaMap github repository: https://github.com/NovaMap-Health/tls-manager-plugin/releases

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

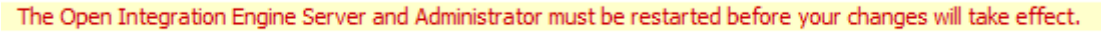


The installed Connectors and Plugins will be listed. At the bottom of the Extensions view is the **Install Extension from File System** section.

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



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



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

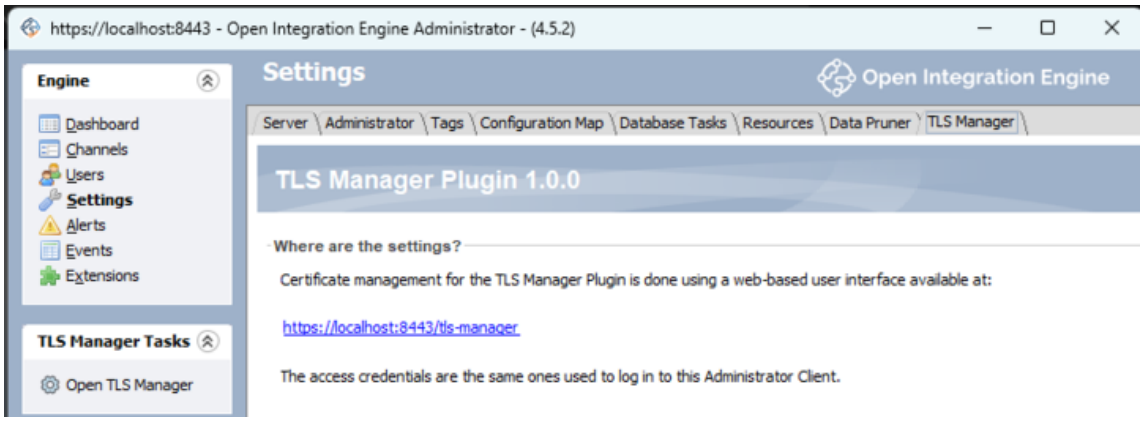
Login

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.

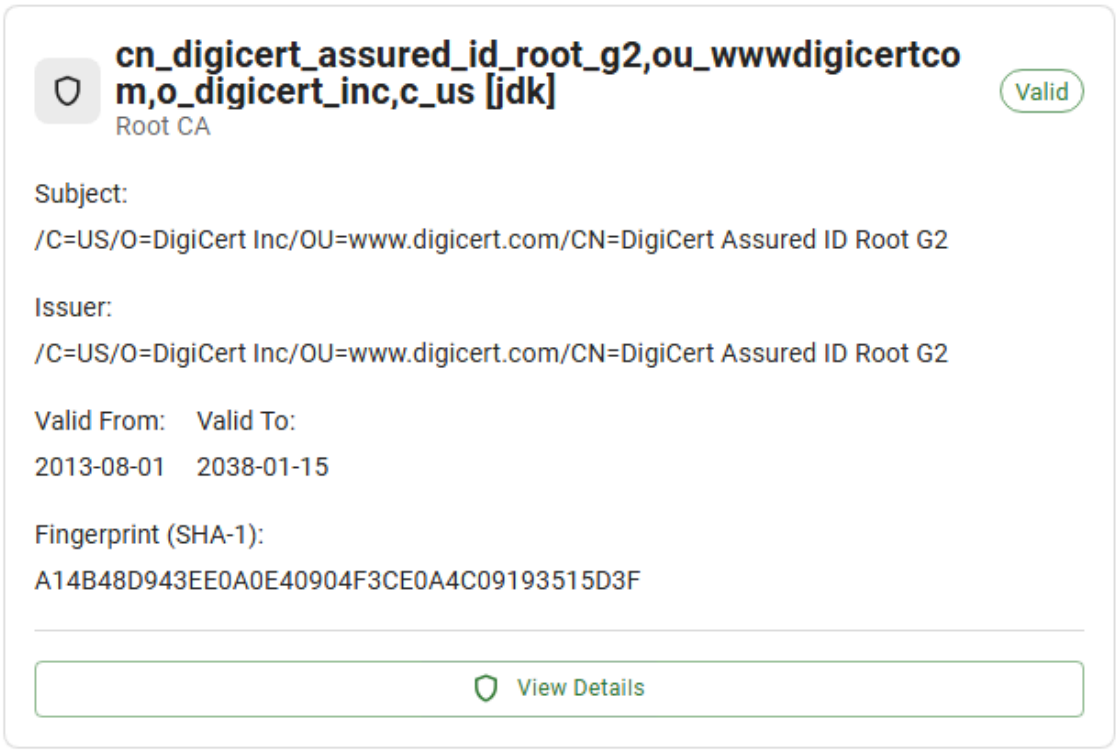
Category	Description	Import Options	Certificate Options
Native Java Certificate Store	Java certificate store.	• None	• View Details
Additional Trusted Certificates	Certificates of remote clients or remote servers.	• Import Certificates • Import From URL	• View Details • Edit Alias • Remove
Local Key Pairs	Certificates with Private Keys. Typically representing the OIE instance owner.	• 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:

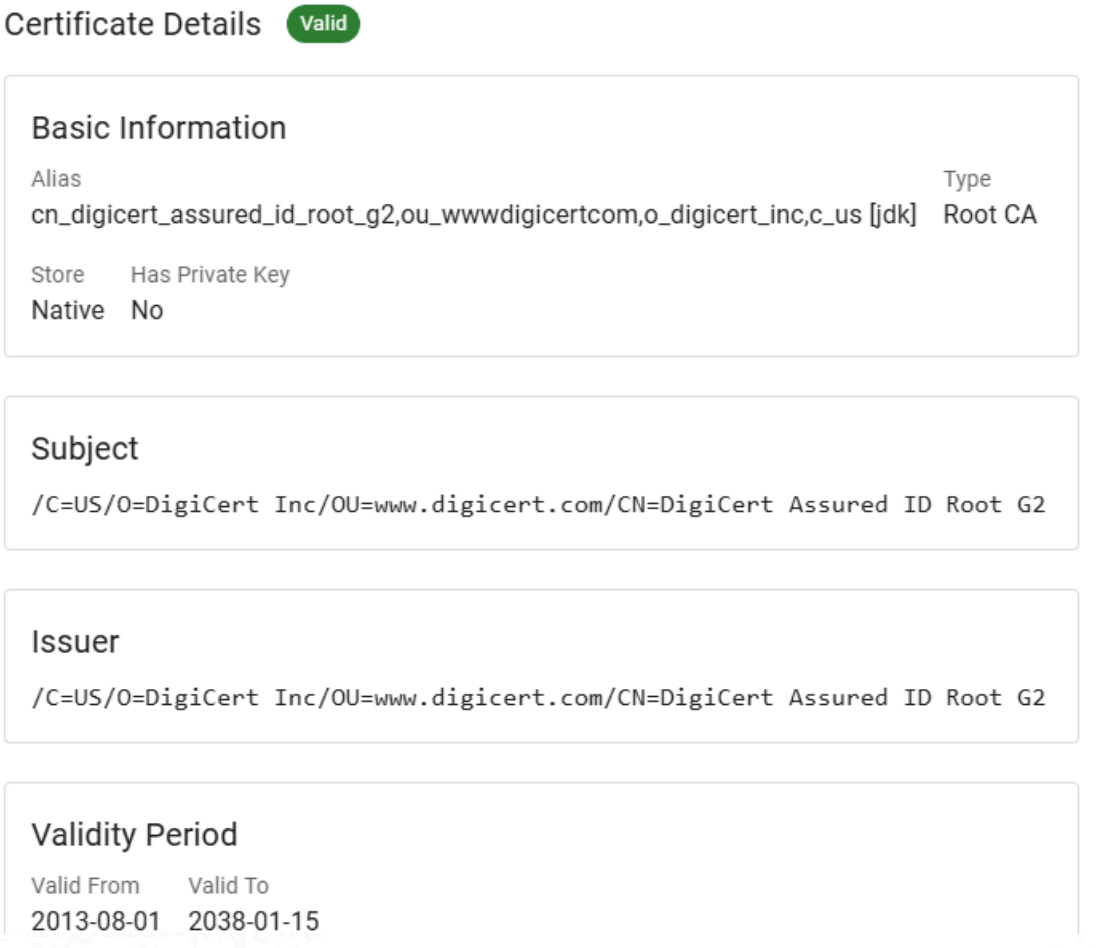


Certificate Details

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
- basicContraints
- keyUsage
- extKeyUsage (if present)
- Channels in Use (if applicable)
- Raw Certificate (Base64)
- Private Key (Base64) (if applicable)
- Certificate Verification

Certificate Details example:



Certificate Details example continued:

Fingerprint

SHA-1

A14B48D943EE0A0E40904F3CE0A4C09193515D3F

Extensions

basicConstraints Critical

keyUsage Critical

digitalSignature, keyCertSign, cRLSign

Raw Certificate (Base64)

-----BEGIN CERTIFICATE----- MIID1jCCAn6gAwIBAgIQC5McOtY5Z+pnI7/Dr5r0SzANBgkqhkiG9w0BAQsFAD
B1 MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUN1cnQgSW5jMRkwFwYDVQQLExB3 d3cuZGlnaWN1cnQuY29tMS
QwIgYDVQQDExtEaWdpQ2VydCBBc3N1cmVkIElEIFJv b3QgRzIwHhcNMTMwODAxMTIwMDAwWhcNMzgwMTE1MTIwMDA
wWjB1MQswCQYDVQQG EwJVUzEVMBMGA1UEChMMRGlnaUN1cnQgSW5jMRkwFwYDVQQLExB3d3cuZGlnaWN1 cnQuY29
tMSQwIgYDVQQDExtEaWdpQ2VydCBBc3N1cmVkIElEIFJvb3QgRzIwggEi MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQDZ5ygvUj82ckmIkzTz+GoeMVSA n61UQbVH35ao1K+ALbkKz3X9iaV9JPrjIgwrvJUXCzO/GU1BBpAAvQxNE
P4Htecc biJVMWWXvdMX0h5i89vqbFCMP4QMls+3ywPgym2hFEwbid3tALBSfK+RbLE4E9Hp EgjAALAcKxHad3A2m
67OeYfcgnDmCXRwVWmvo2ifv922ebPynXApVfSr/5Vh88lA bx3RvpO704gqu52/clpWcTs/1PPRCv4o76Pu2ZmvA9
OPYLfykqGxvYmJHzDNw6Yu YjOuFgJ3RFrngQo8p0Quebg/BLxcoIfhG69Rjs3sLPr4/m3wOnyqi+RnlTGNAgMB AA
GjQjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgGGMB0GA1UdDgQW BBTOw0q5mVXyuNtgv6l+vVa1lza
n1jANBgkqhkiG9w0BAQsFAAOCAQEAvgVVjOPI QW5pJ6d1Fe88hi7v0p3GeDgda7ajkmkuQGvhfQTUiaWxMTeKvSHM

Certificate Verification

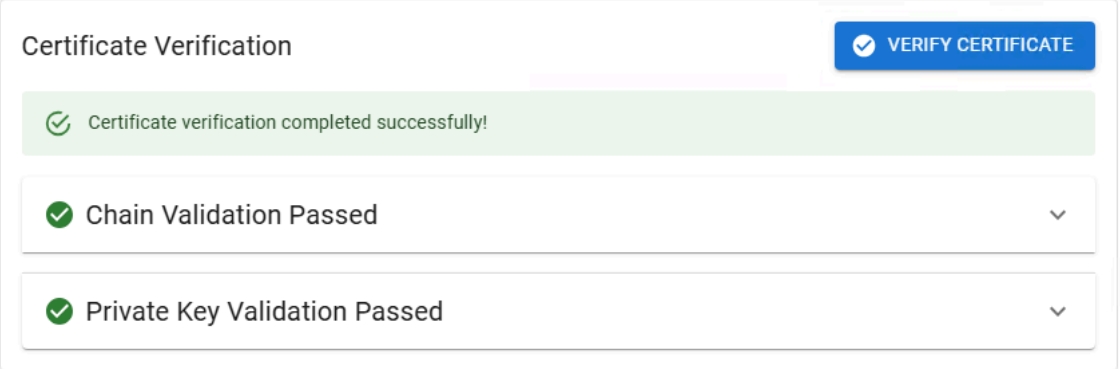
VERIFY CERTIFICATE

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:



5. TLS Manager: Importing

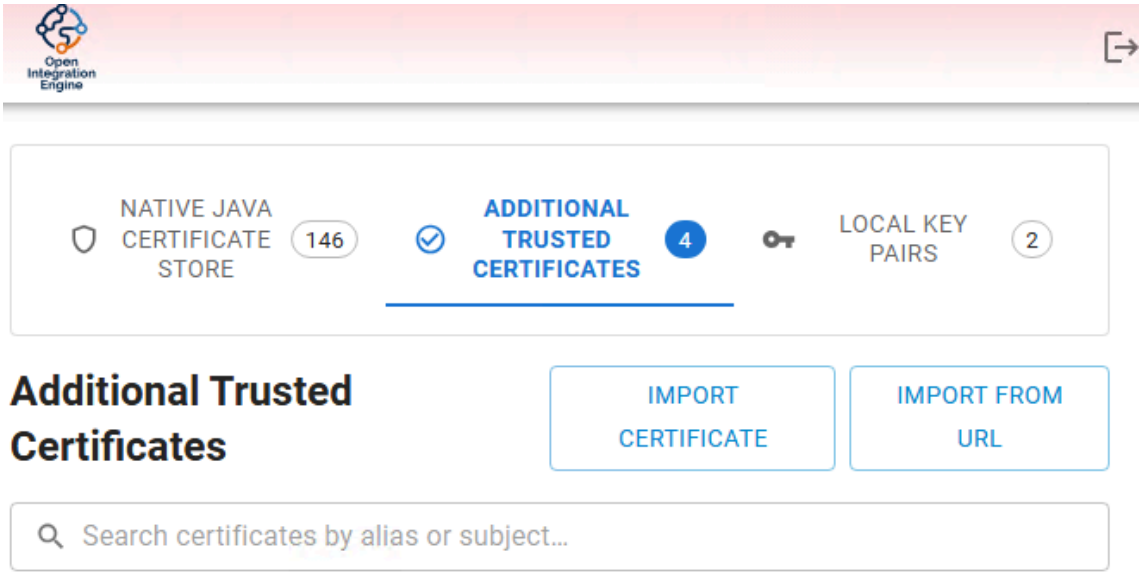
Additional Trusted Certificates

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)

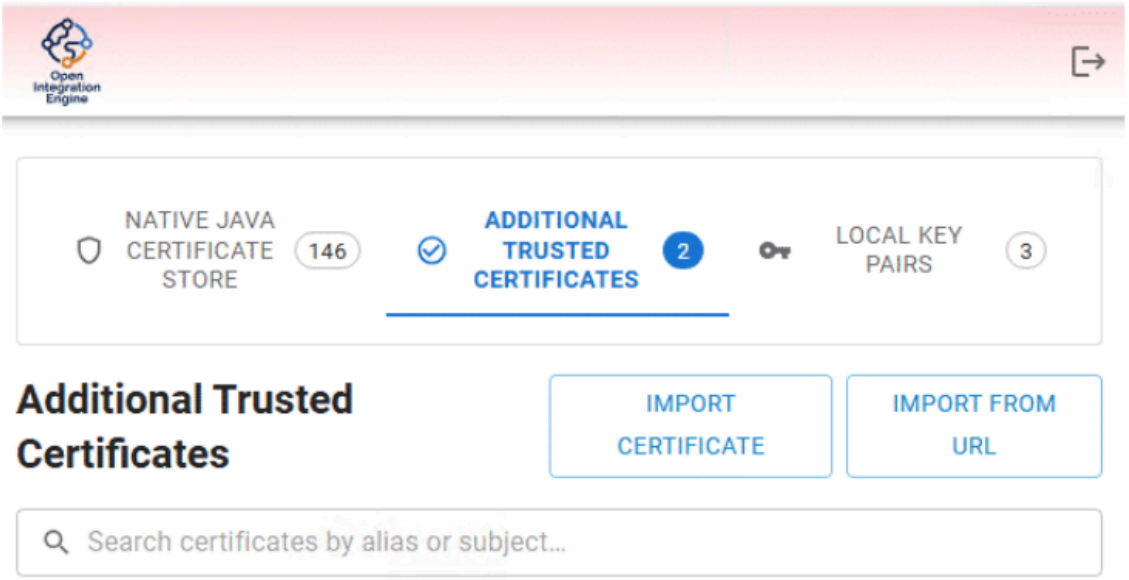
They can also be used by these connector types as **Trusted Client Certificates**:

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

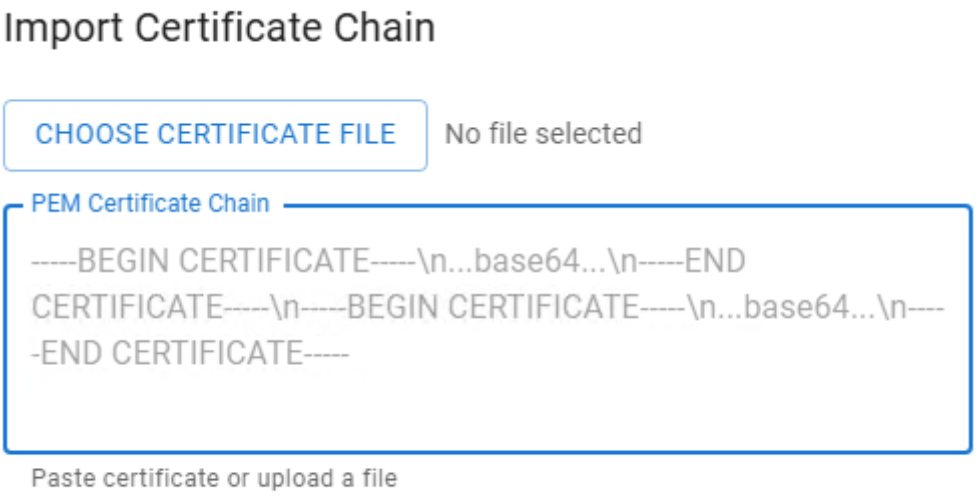


Import Method 1: Choose Certificate File

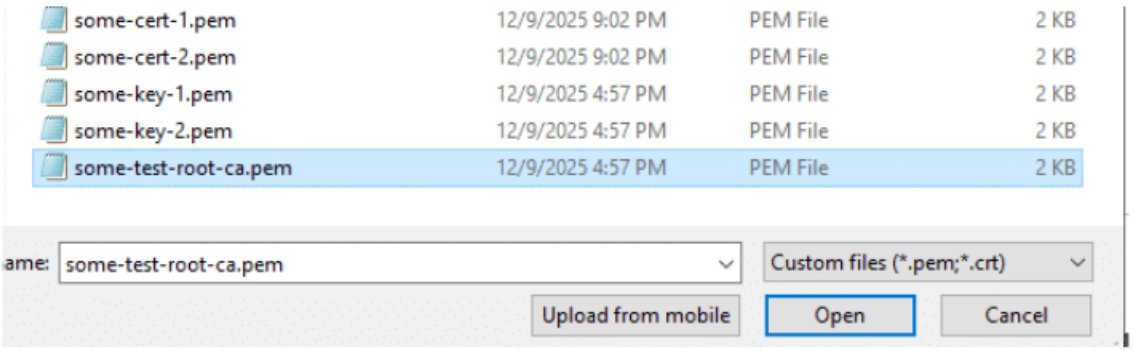
To import a certificate from file click the **Import Certificate** button:



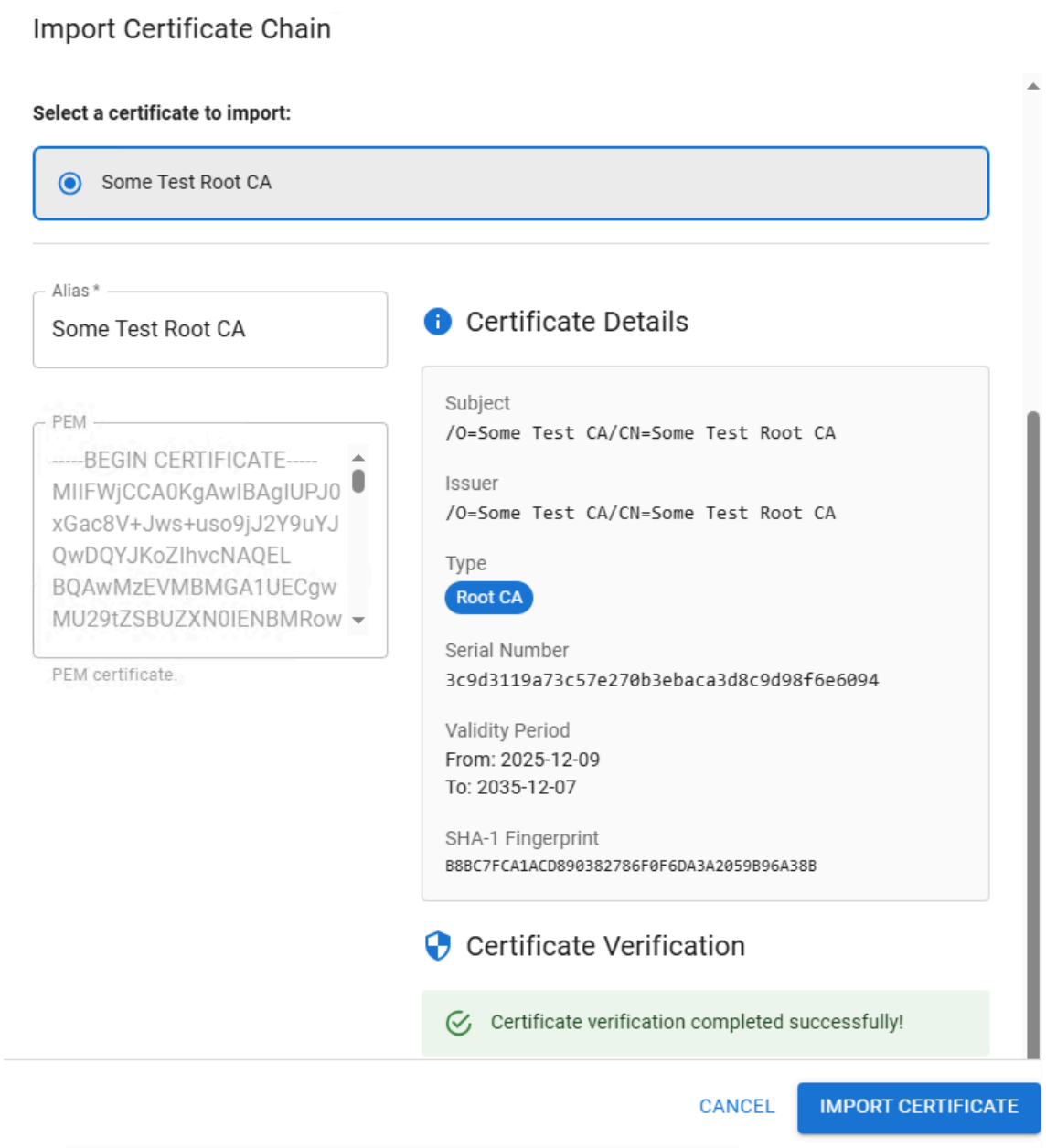
Click **Choose Certificate File:**



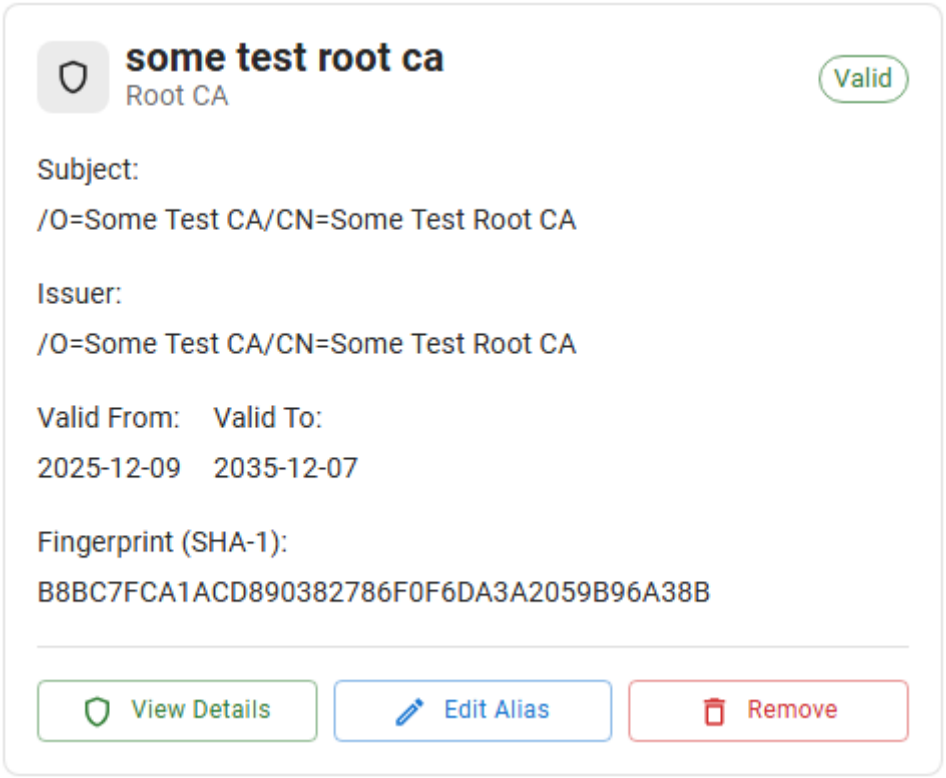
Select your certificate. The certificate file name on disk must end in .pem or .crt:



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



The certificate after successful import:

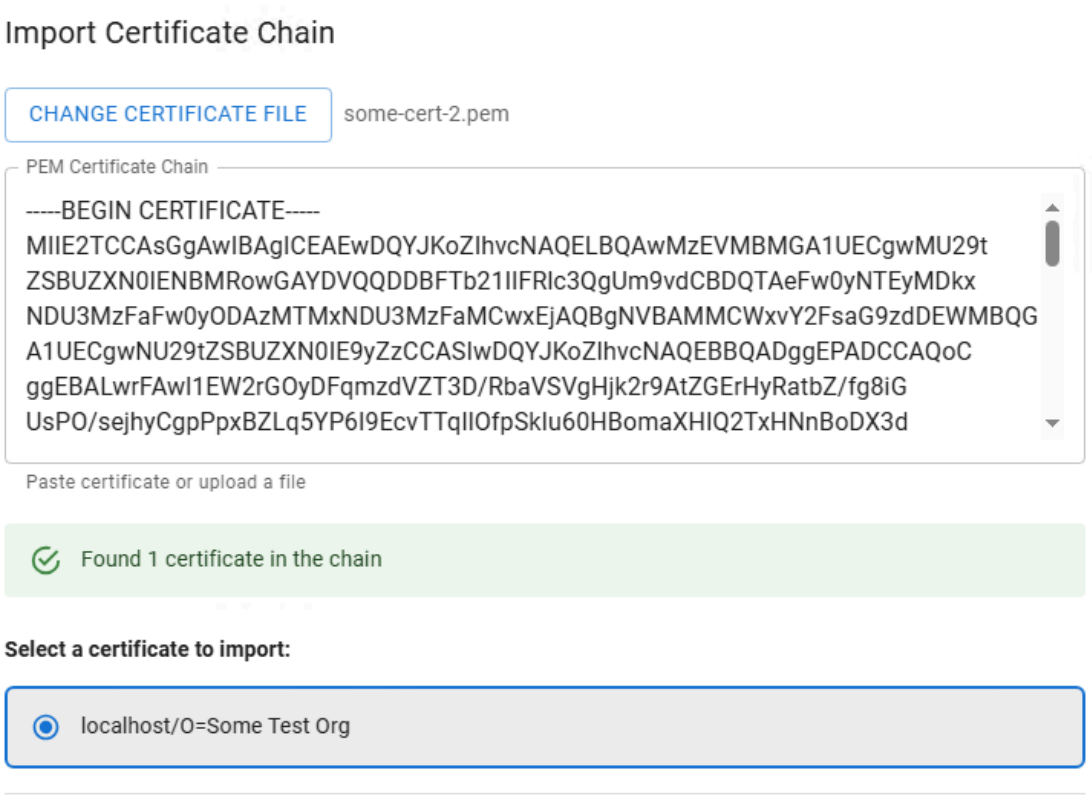


Import Method 2: Paste Certificate

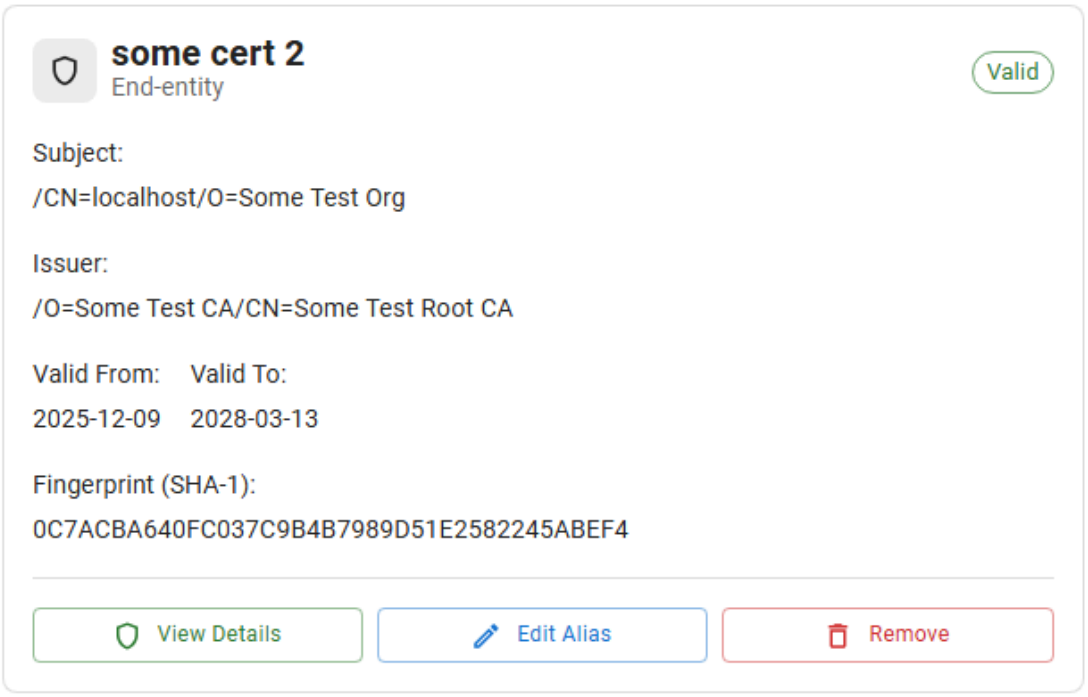
To paste a certificate, first click the **Import Certificate** button:

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

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



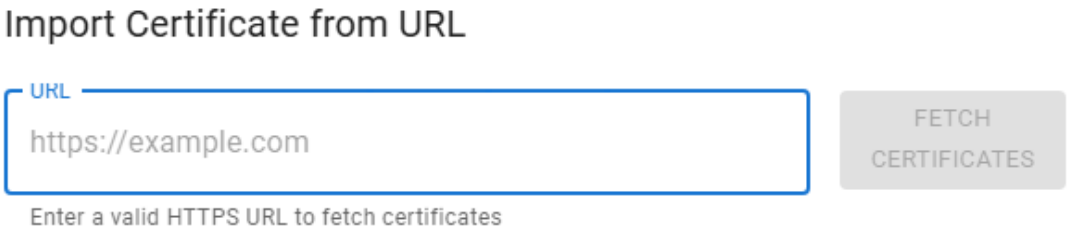
The imported certificate:



Import Method 3: Import From URL

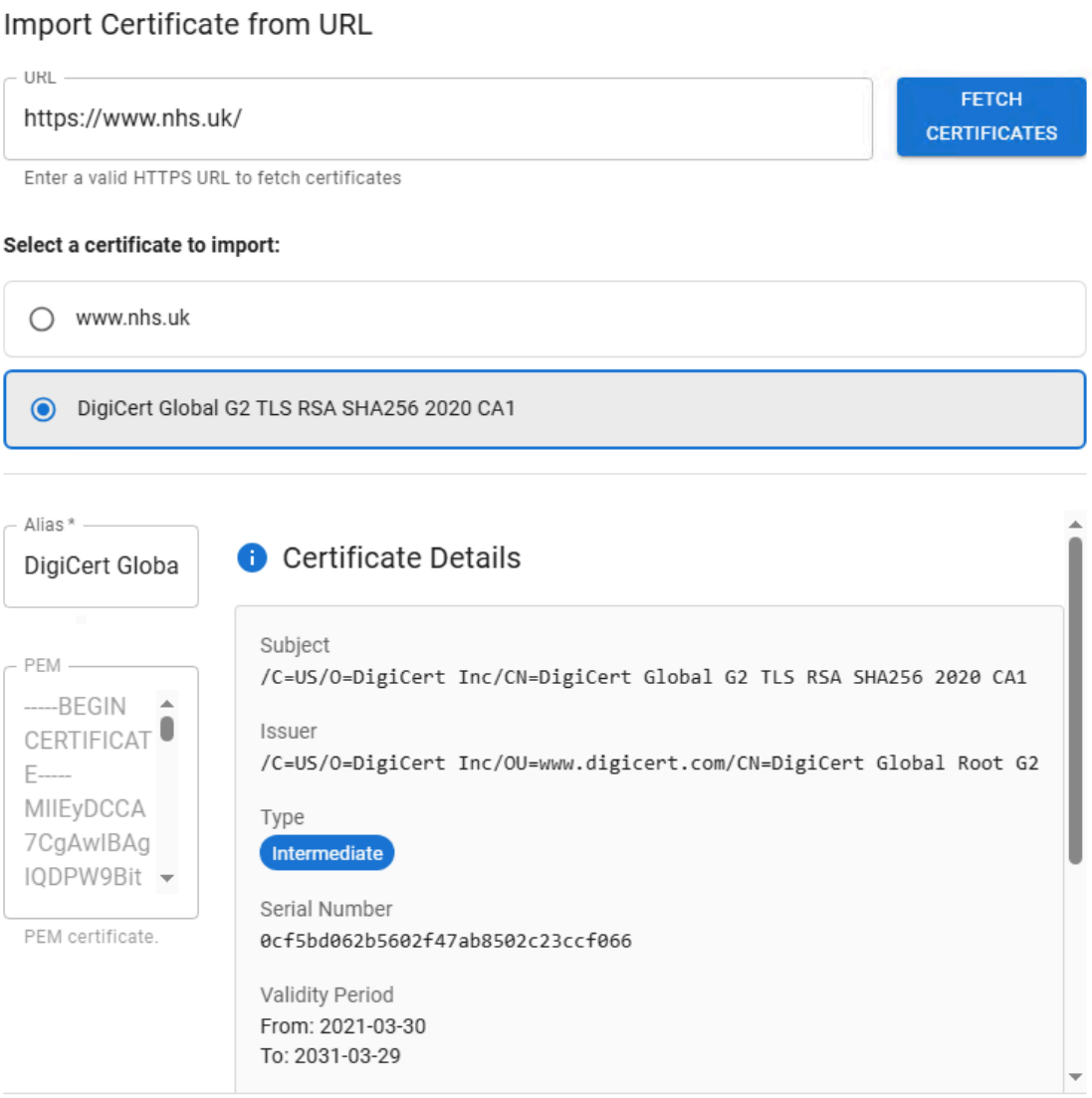
To import a certificate from a URL click the **Import From URL** button:

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

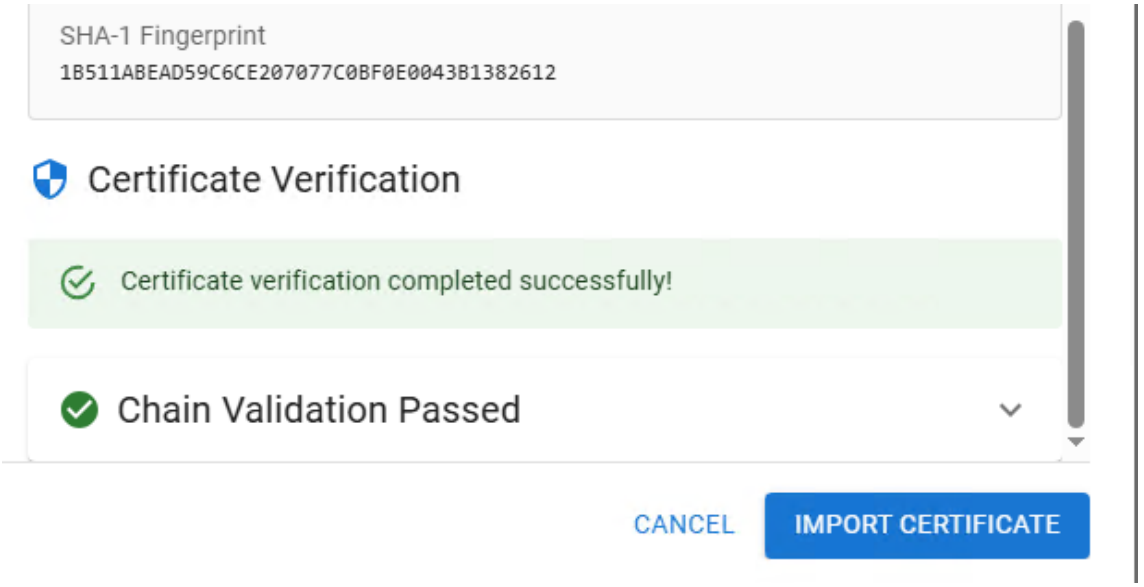


Click **Fetch Certificates**.

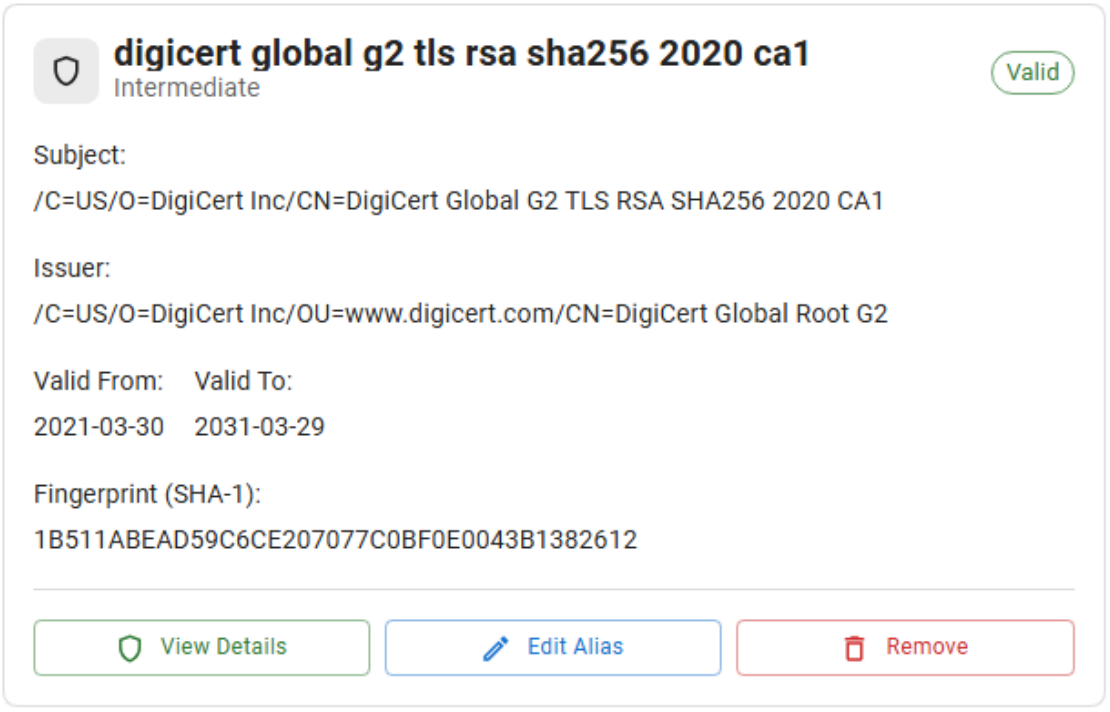
The URL may offer more than one certificate. Select one:



Click **Import Certificate**:



The imported certificate:



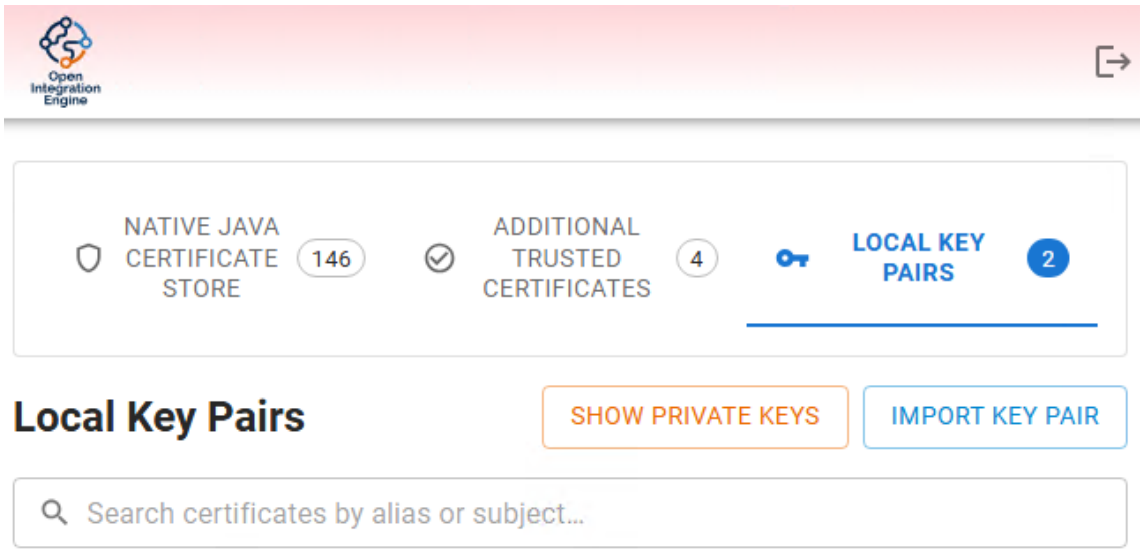
Local Key Pairs

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)



Import Key Pair

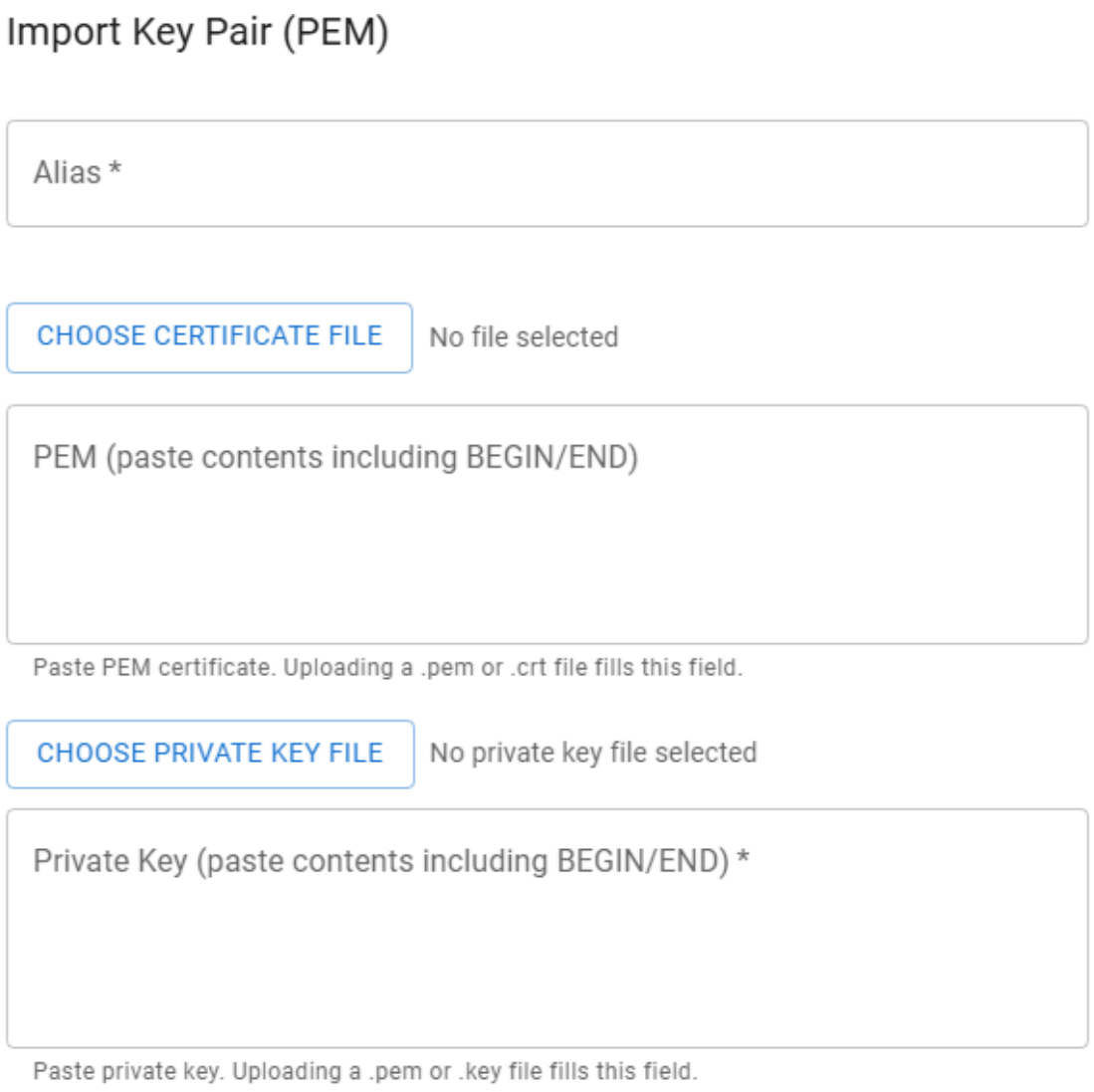
To import a local certificate click **Import Key Pair.**

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

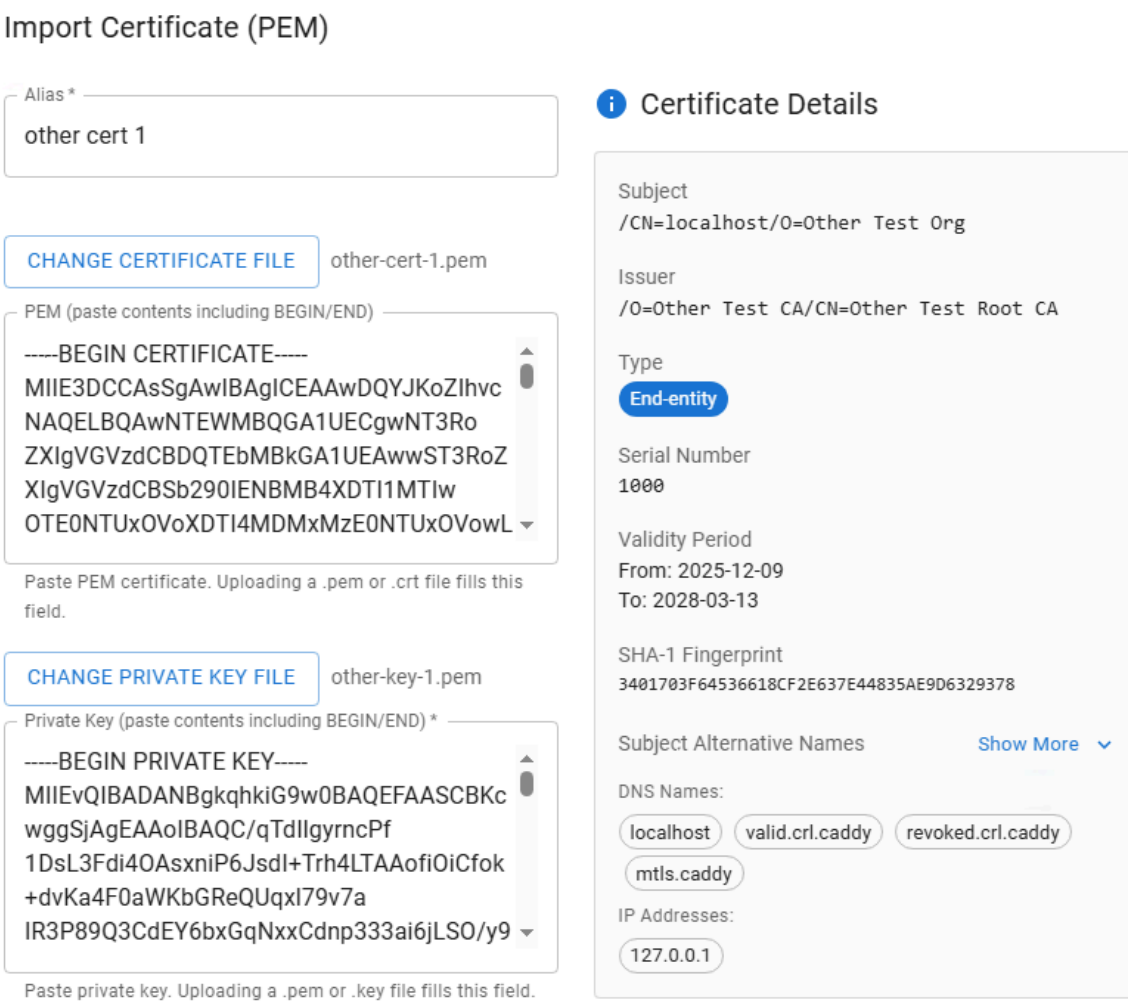
Click **Choose Certificate File** and select the public key.

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

Only .pem or .crt files are accepted.

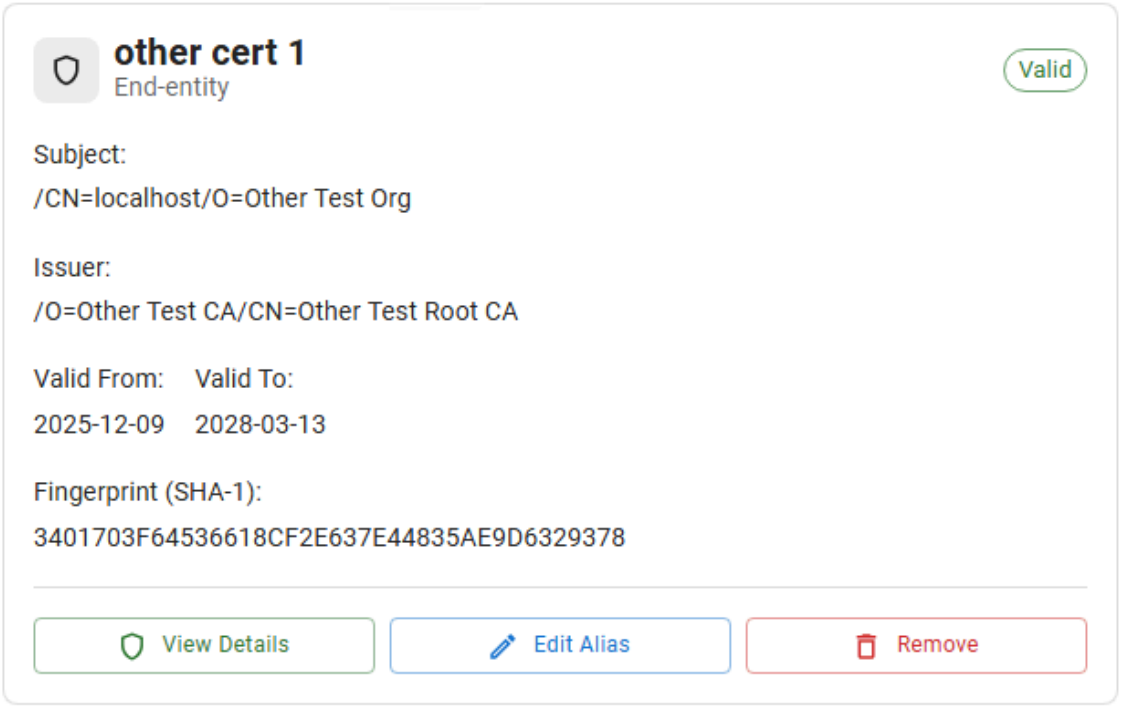


In this example we have selected a key pair and set an alias:



Click **Import Certificate**.

The imported certificate:

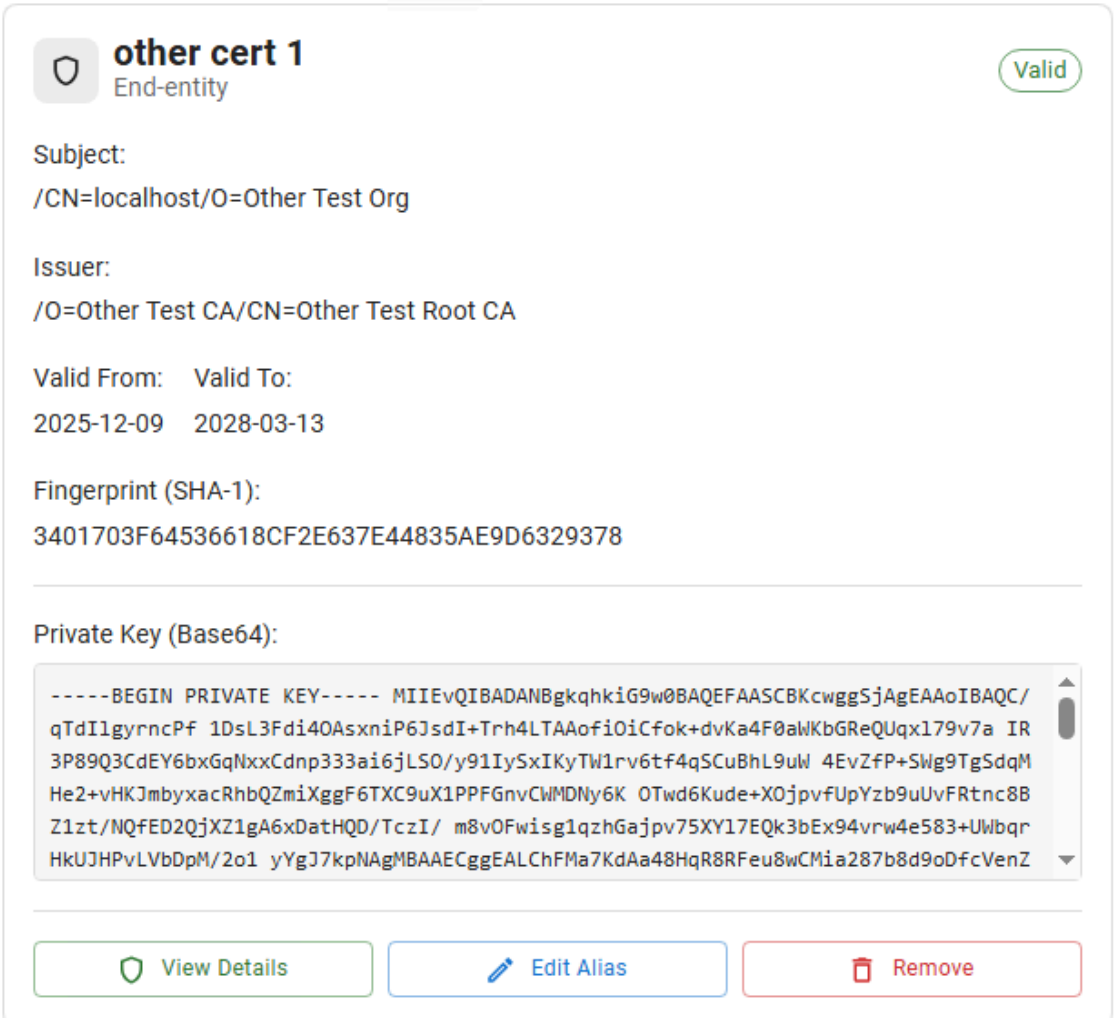


Show Private Keys

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

SHOW PRIVATE KEYS

For example:



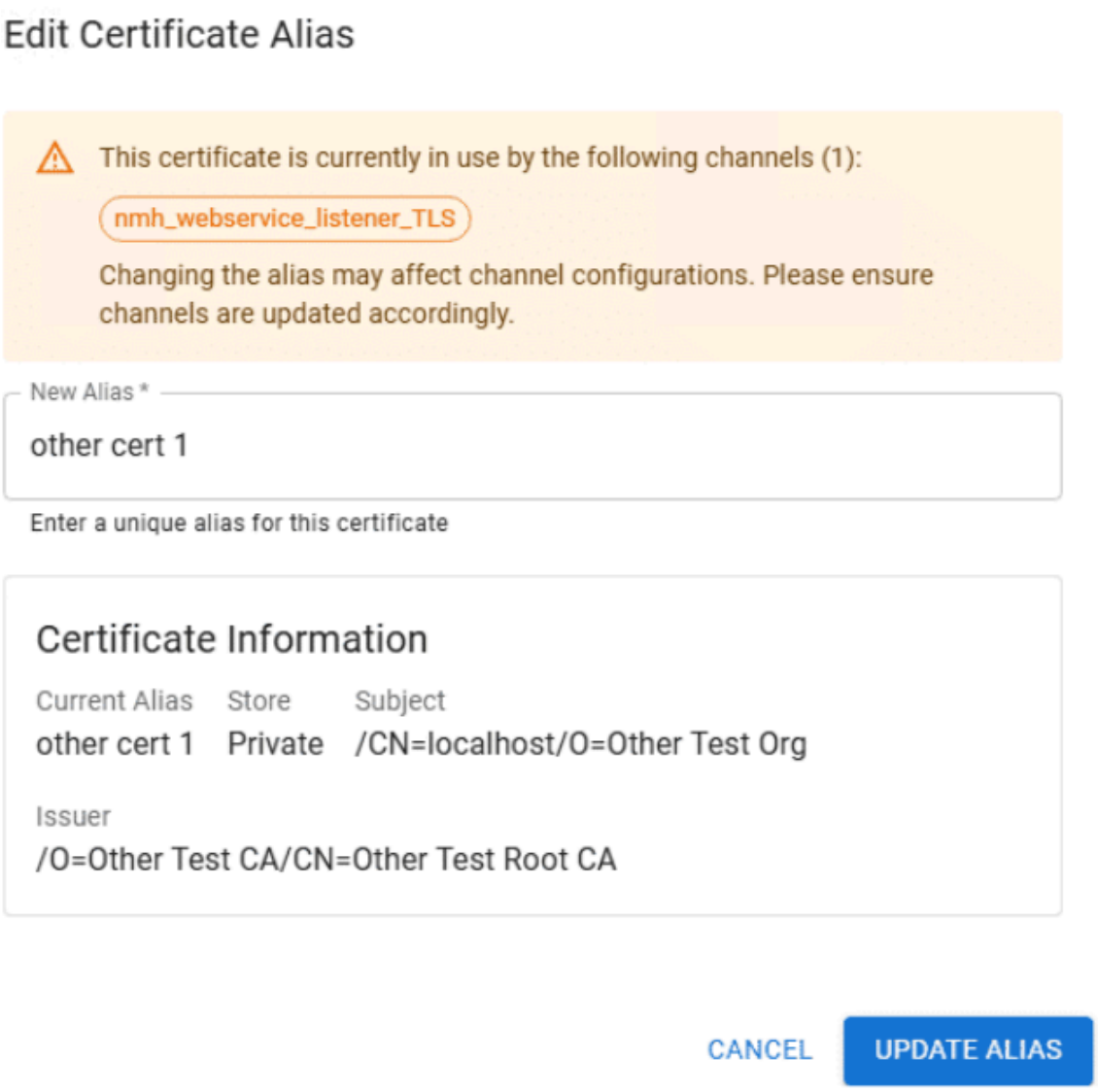
6. TLS Manager: Common Tasks

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.

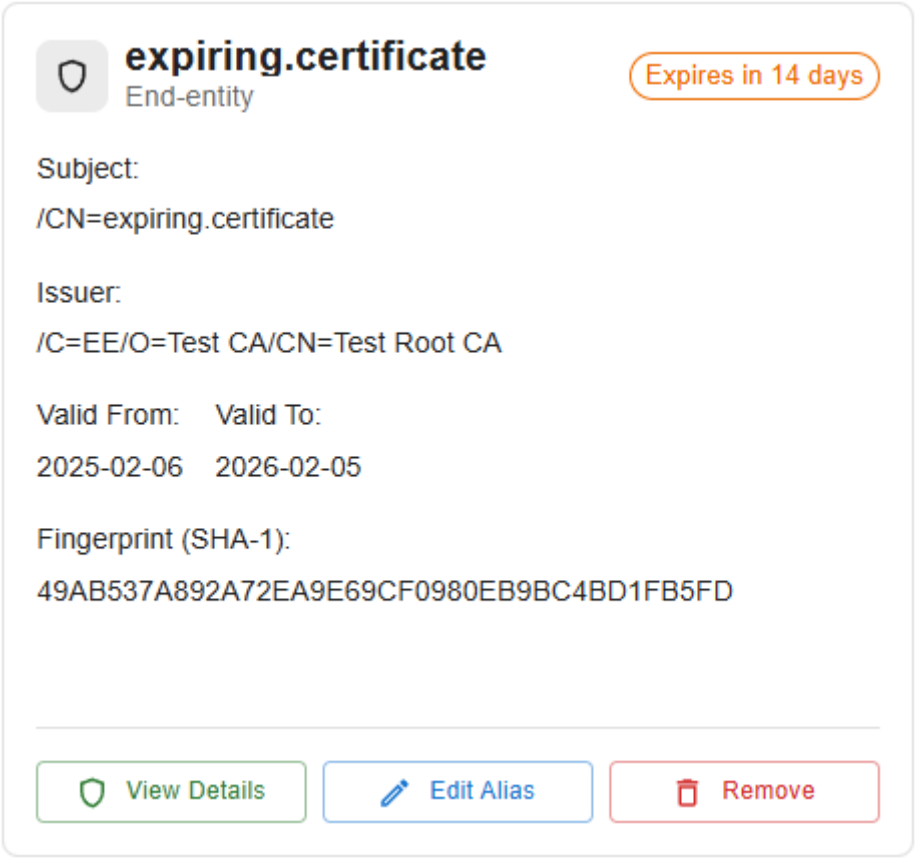
If the certificate is used by a channel under its alias, a warning will be displayed:



To update click Update Alias.

Expiry Warning

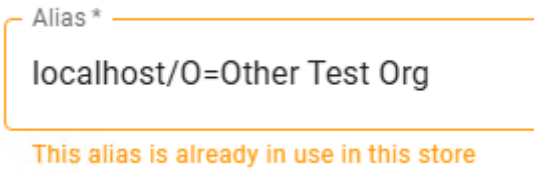
When a certificate is within 30 days of its expiry date you will see a warning indicating how many days are remaining.



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:

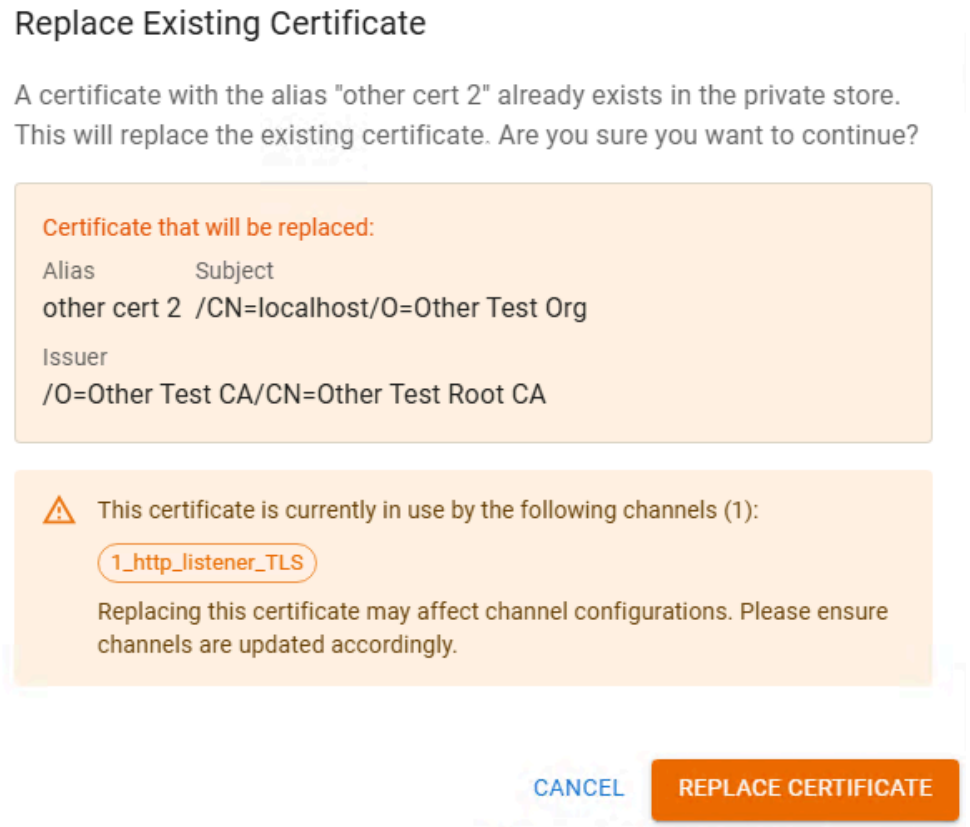


If you continue with certificate import you will be advised that you will be replacing the certificate with the same Alias.

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**:

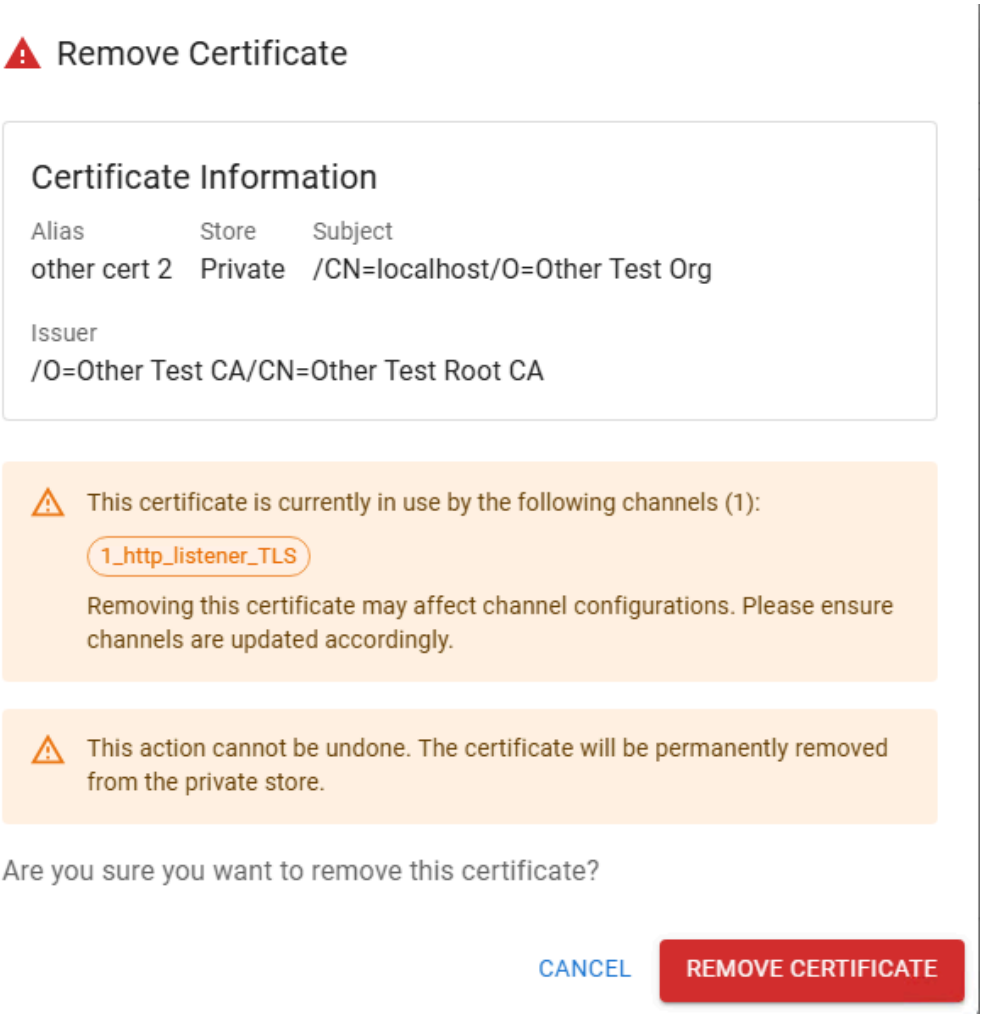


Certificate Removal

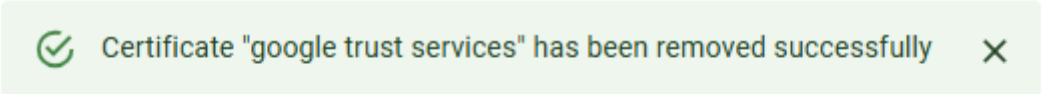
To remove a certificate click **Remove**.

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**:



Once removed the successful removal notification will appear:

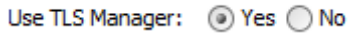


7. Sender TLS Settings

These settings are available to the following connector types:

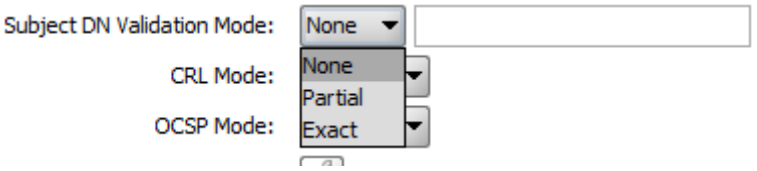
- **HTTP Sender**
- **Web Service Sender**
- **TCP Sender**

Use TLS Manager: *Yes/No*



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*



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

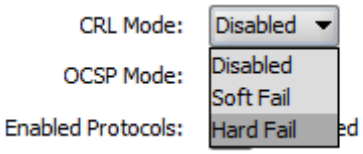
Exact: Distinguished Name must match the specified RDN:

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

Successful match:

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

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

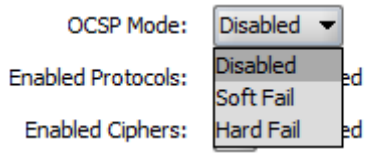


To have the endpoint certificate automatically checked against a Certificate Revocation List (CRL) service.
The service used is defined within each certificate. It will typically be the Issuing Authority.

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*

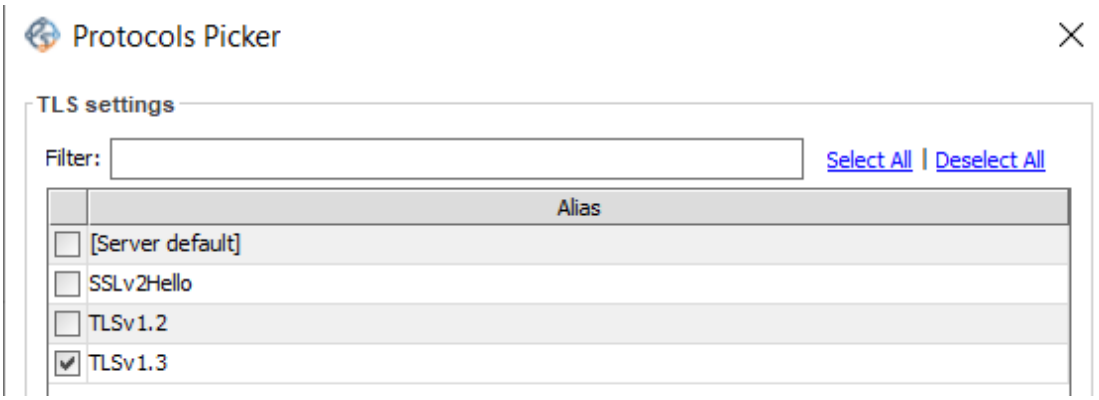


To have the endpoint certificate automatically checked against an Online Certificate Status Protocol service (OSCP).
The service used is defined within each certificate. This is typically the Certificate Authority (CA) that issued it.

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.

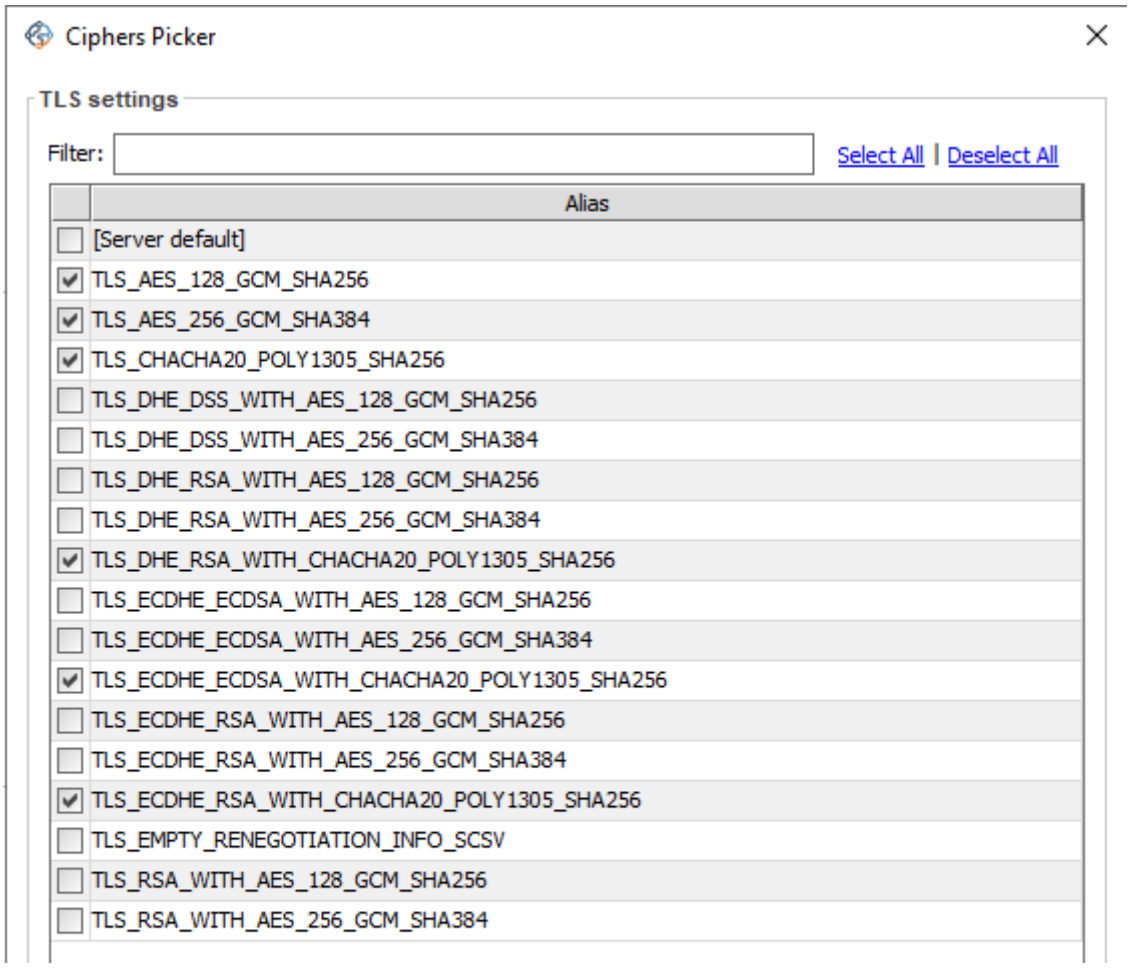
Enabled Protocols:



Select the acceptable protocol or protocols.

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

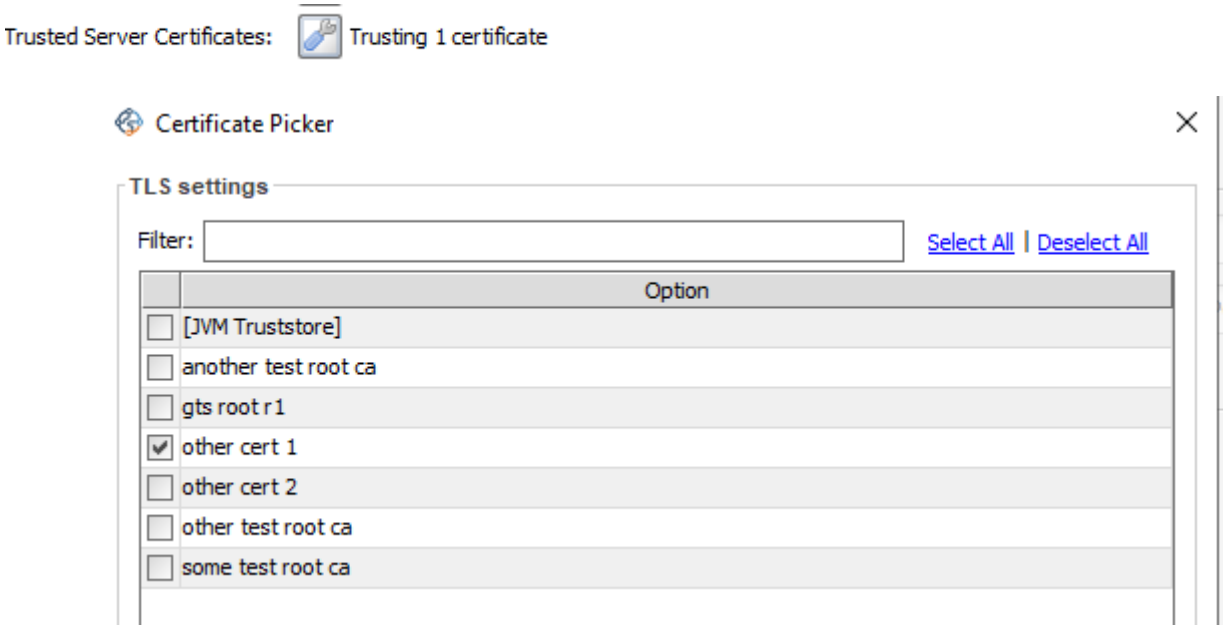
Enabled Ciphers:



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

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

Trusted Server Certificates:



Enabled for TCP Sender in Client Mode only.

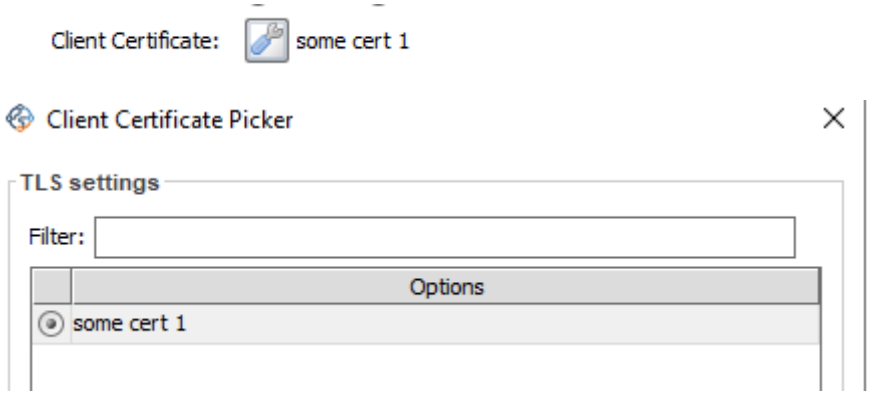
The trusted server. Select from **Additional Trusted Certificates.**

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:

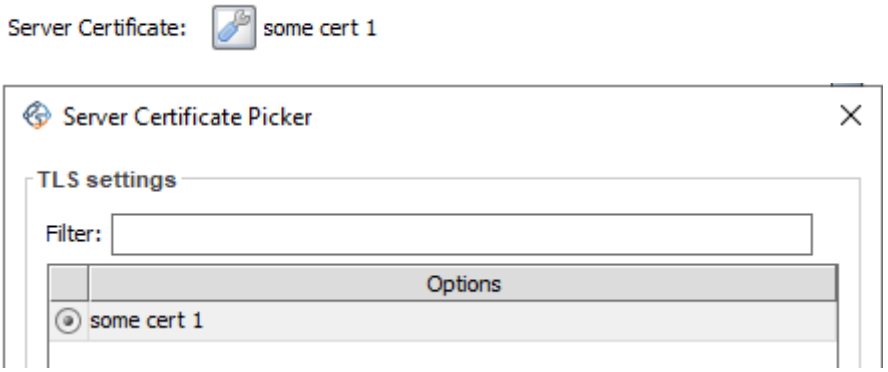


Enabled for TCP Sender in Client Mode only.

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:

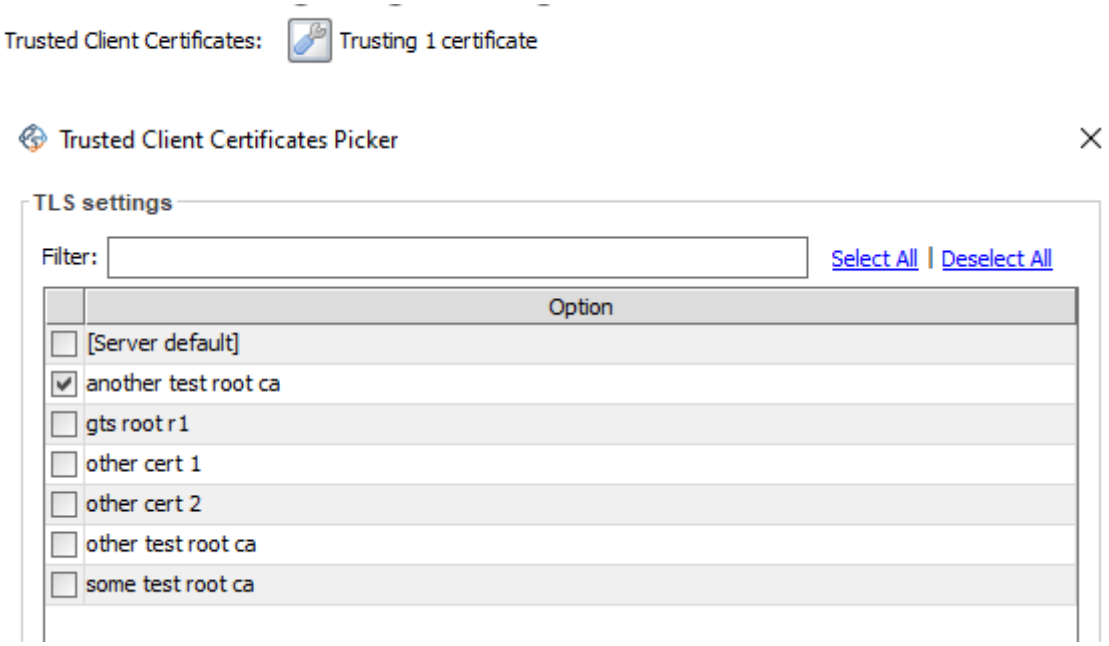


Your server certificate. Select from **Local Key Pairs**.

Client Authentication Mode:

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

Trust Client Certificate:



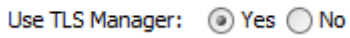
The certificate of the entity acting as the client (mTLS). Select from **Additional Trusted Certificates.**

8. Listener TLS Settings

These settings are available to the following connector types:

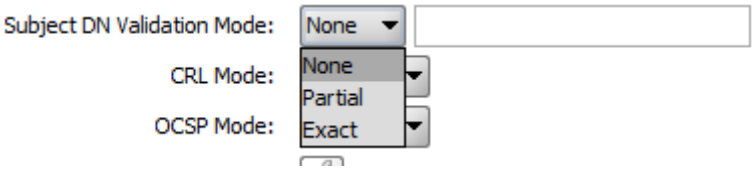
- **HTTP Listener**
- **Web Service Listener**
- **TCP Listener**

Use TLS Manager: *Yes/No*



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*

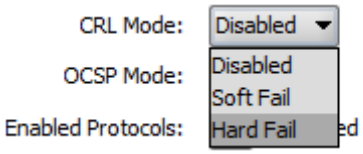


A way of filtering using Relative Distinguished Names (RDN).

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*

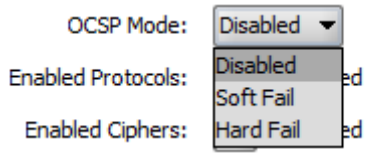


To have the endpoint certificate automatically checked against a Certificate Revocation List (CRL) service.
The service used is defined within each certificate. It will typically be the Issuing Authority.

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*

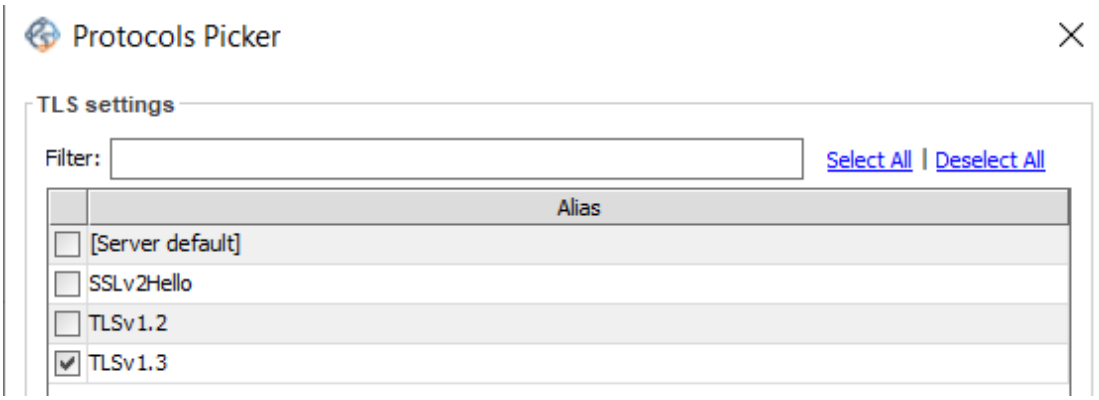


To have the endpoint certificate automatically checked against an Online Certificate Status Protocol service (OSCP).
The service used is defined within each certificate. This is typically the Certificate Authority (CA) that issued it.

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.

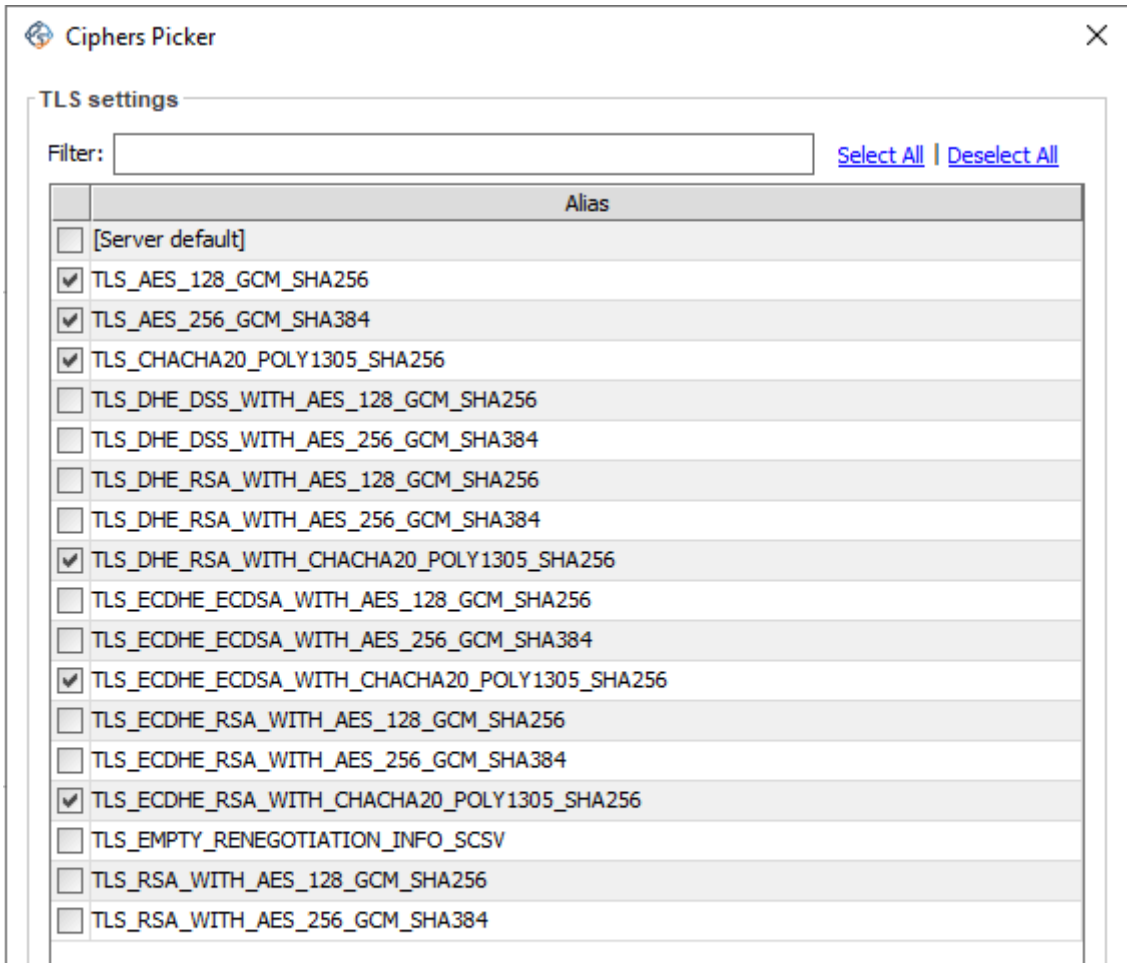
Enabled Protocols:



Select the acceptable protocol or protocols.

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

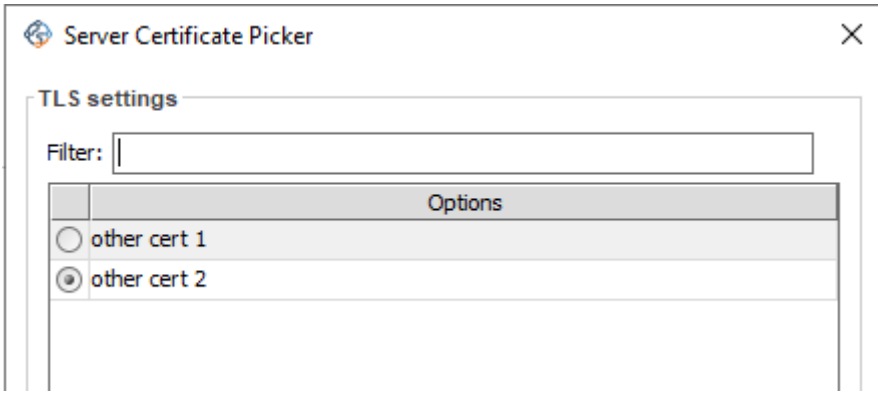
Enabled Ciphers:



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

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

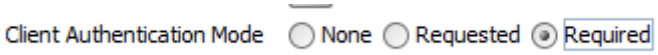
Server Certificate:



Enabled for TCP Listener in Server Mode only.

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

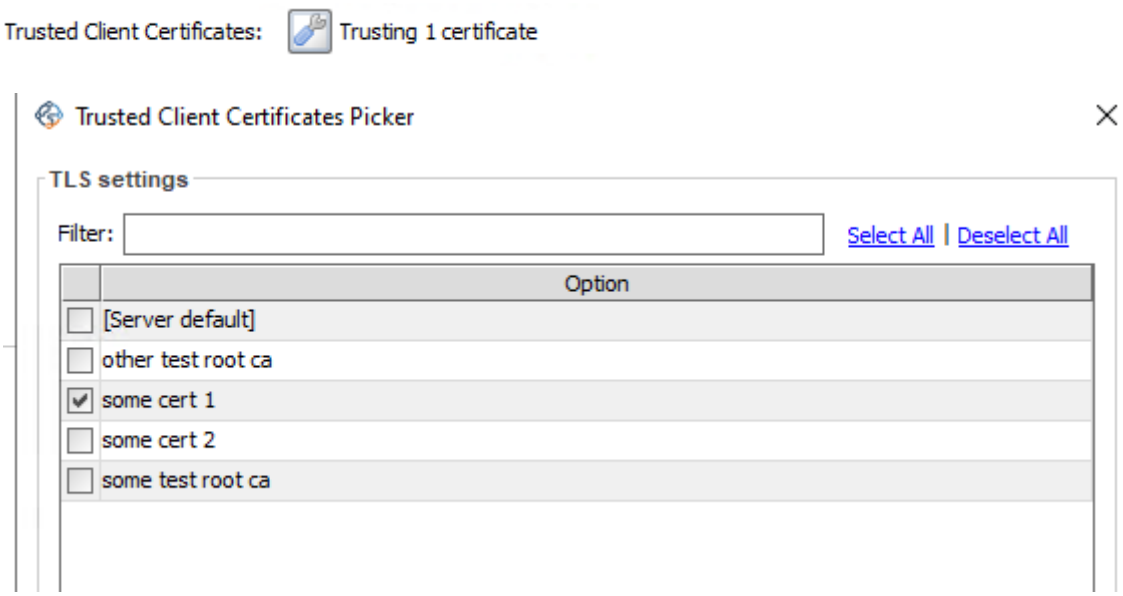
Client Authentication Mode: *None/Requested/Required*



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:

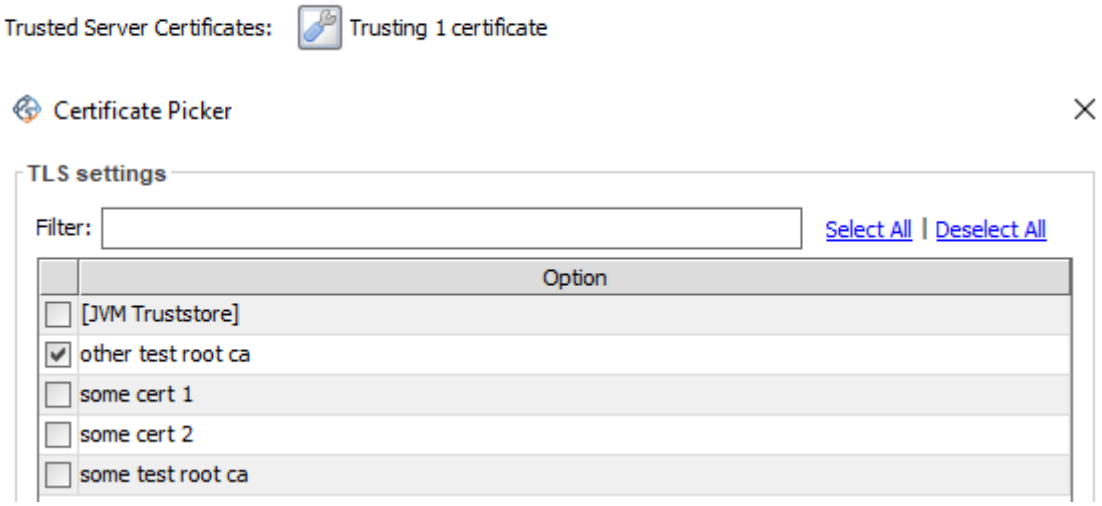


Enabled for TCP Listener in Server Mode only.

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:

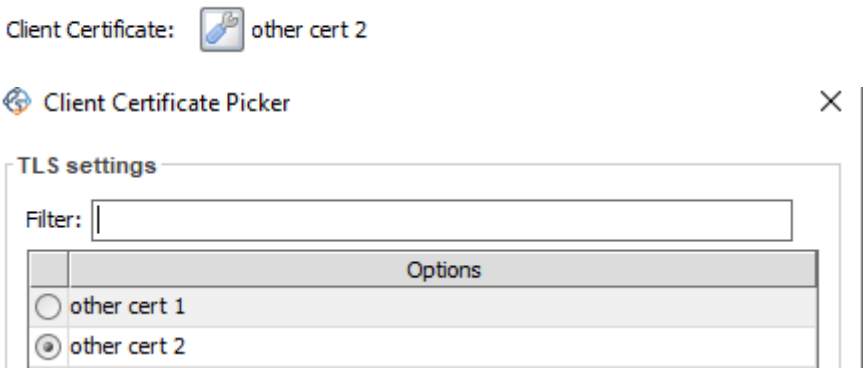


The certificate associated with the server. Select from **Additional Trusted Certificates.**

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))

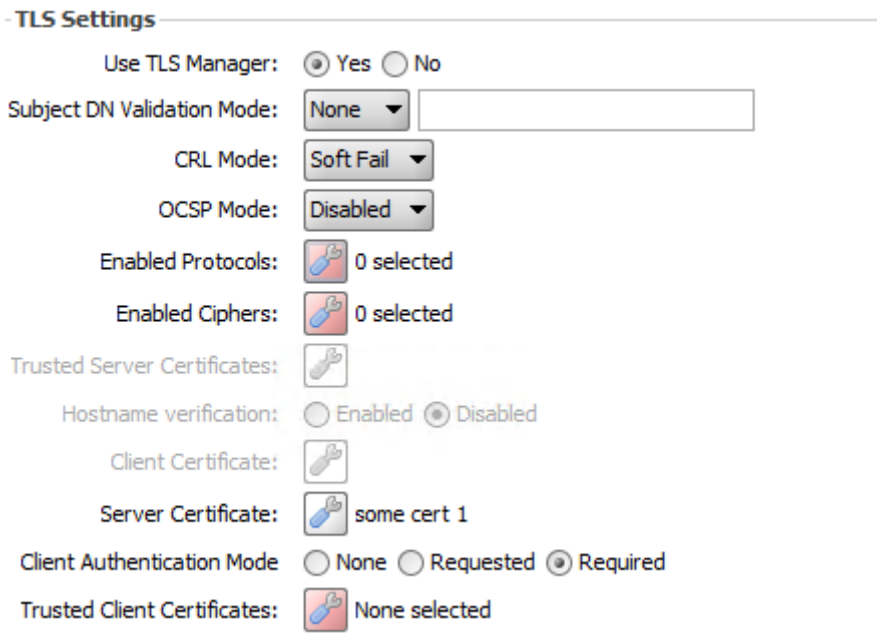
Client Certificate:



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:



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:

API Calls

POST /tlsmanager/_cacheWsdlFromUrl

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/_isWsdlCached

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