Technical Document

Niagara Signing Service Guide

niagara

Signing Service

- **[Overview (Signing Service)]**
  The Signing Service was added in Niagara 4.13, a service to which components within running stations may securely submit Certificate Signing Requests (CSR). This is performed to obtain signed X509 certificates and to send additional requests for renewal of these certificates before they expire. The Signing Service fulfills these requests by returning certificates signed by a designated Certificate Authority (CA).
- **Signing Service high level types**
  The following section describes the key parts of the Signing Service.
- **Signing Service high level workflow**
  This example assumes the usage of the **Fox Signing Transport** component.
- **Installing a Signing Requester**
  There are different ways to add the Signing Requester to station components.

Overview (Signing Service)

The Signing Service was added in Niagara 4.13, a service to which components within running stations may securely submit Certificate Signing Requests (CSR). This is performed to obtain signed X509 certificates and to send additional requests for renewal of these certificates before they expire. The Signing Service fulfills these requests by returning certificates signed by a designated Certificate Authority (CA).

Use:

Any component that requires a signed certificate to fulfill its functionality may be suitable to utilize the Signing Service. This could be a component that uses a certificate as way of authentication with an external service or protocol. It could also be a component that accepts connections from outside clients and uses a server certificate to prove identity. For a list of components that can currently utilize the Signing Service, see "Supported components" below.

Benefits:

- Using the Signing Service removes the effort of manually generating individual X509 certificates specific for each component that requires one, and the additional task of manually signing that certificate with a CA.

- Once components are onboarded with the Signing Service, they will automatically request new certificates prior to renewal to save time on having to repeatedly recommission a fleet of controllers.

- It promotes good security practice of shorter certificate expiry periods.

This service is tailored to the need for stations to generate and renew one of many types of certificate on a rolling basis.

- **Supported components**
  The following list contains components that currently can utilize the Signing Service.
- **Signing Service — FAQs**

Parent topic: Signing Service

Supported components

The following list contains components that currently can utilize the Signing Service.

Module	Component	Usage	As of release
abstractMqttDriver	Aws Jitp Mqtt Authenticator	Requests client SSL certificates to enable connection to Amazon Web Services IoT platform via MQTT.	Niagara 4.13
fox	Fox Service	Requests a server SSL certificate that is used when establishing connections with clients, such as Workbench, or other Niagara stations.	Niagara 4.14
web	Web Service	Requests a server SSL certificate that is used by the station's Web Server, when establishing connections with web clients, such as browsers, or third-party web service clients.	Niagara 4.14
web	additional Https Cert	Requests a server SSL certificate for an additional Https Certificate that may be used by the station's Web Server in use cases such as SNI where multiple certificates are being presented by	Niagara 4.14

		the server.	
platform	Niagara platform	Requests a server SSL certificate that is used for establishing connections with Workbench clients, and any remote components that connect to the platform such as provisioning jobs, or Niagara Network connections.	Niagara 4.14
abstractMqttDriver	Generic Mqtt Authenticator	Requests a client SSL certificate that is used for mutual SSL authentication with Mqtt brokers	Niagara 4.14

Parent topic: Overview (Signing Service)

Signing Service — FAQs

Q: My Web Service has signed certificates, however my browser still shows a warning on connection to the station?

A: Ensure that the CA certificate that signed your Web Service certificate exists in the Trusted Certificate Authorities store on your browser's host OS. Also confirm that the certificate's Subject Alternative Name extension contains a value that matches the host/domain name or IP address as it appears in the browser address bar.

Q: I have an existing self-signed certificate for my component/service. May I use it with the Signing Service to get it properly signed?

A: Yes, if you haven't already, import the existing certificate into the User Key Store, and then set the alias and correct password in the Cert To Sign And Apply property of your component's **Signed Cert Config**.

Q: My signing requester has an existing signed certificate, however instead of renewal, I wish to replace the existing cert with a newly generated certificate. Alternatively, I successfully onboarded a signed certificate, but now I have made configuration changes (to the profile or requester) and wish to replace the existing certificate with a newly generated one with the new settings applied. How do I do this?

A: You may simply change the alias/password for the certificate to sign and re-invoke the **Onboard** action. If you wish to retain the same alias, do the following:

- Delete the certificate with the existing alias from the User Key Store on the requesting station.
- From the Certificate Store (within the relevant Signing Profile on the station that contains the Signing Service), delete the record with the matching Requester Id.

Q: My signed certificates are due to expire. What do I need to do?

A: Components onboarded with the Signing Service will automatically renew the certificates (by default) when 8% of the certificate's valid period is remaining. The Fox/Web Services will restart and begin using the renewed certificate immediately. Other components may use the new certificate when a current connection is broken and restored. It would be advisable to check that the CA certificate on the relevant Signing Profile has not expired prior to renewal.

Q: My CA certificate is due to expire. What do I need to do?

A: If your new CA certificate will use the same alias as before, simply import the new CA into the User Key Store on which the Signing Service is running. If the alias has changed, you will additionally need to change the CA alias property in the relevant Signing Profiles.

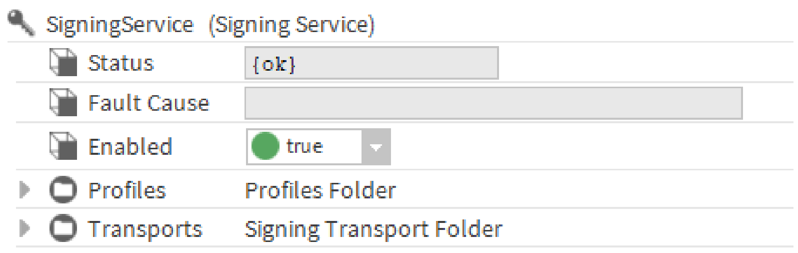
If the CA has been used to sign the certificate for the Fox Service on a Supervisor prior to upgrade, install the new CA into the User Trust Store on all remote stations that make Niagara Network connections back to the Supervisor to achieve a seamless transition. A Supervisor Provisioning Job would be ideal in this situation. Additionally, install to the User Trust Store of Workbench installations that connect to that Supervisor to avoid connection warnings.

If the CA has been used to sign the certificate for the Web Service on a Supervisor prior to upgrade, install the new CA into the Trusted Certificate Authorities store on the host OS of any browsers that connect to ensure the connection is trusted by the browser.

Parent topic: Overview (Signing Service)

Signing Service high level types

The following section describes the key parts of the Signing Service.

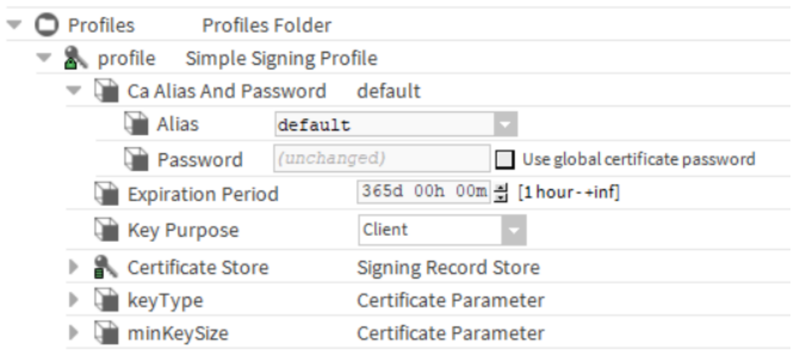


SigningService (Signing Service)

Status	{ok}
Fault Cause	
Enabled	● true
Profiles	Profiles Folder
Transports	Signing Transport Folder

Profiles

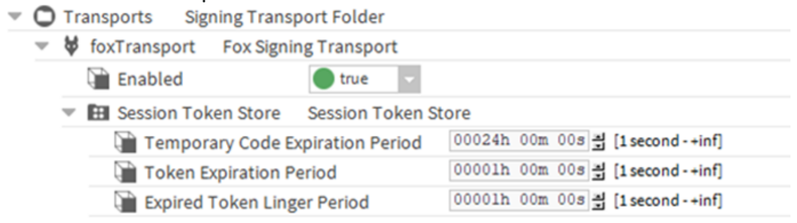
Profiles are where you define the alias and a unique or global certificate password of the CA certificate that will be used to sign all Certificate Signing Requests (CSR) associated with that profile. In addition, you may define values for various certificate fields that are applied to the signed certificate, and to validate those fields within the CSR supplied by the remote station. Each profile also holds a Certificate Store where you may view the records of the CSR associated with that profile, including the signed certificate.



Profiles	Profiles Folder	
profile	Simple Signing Profile	
Ca Alias And Password	default	
Alias	default	
Password	(unchanged)	Use global certificate password
Expiration Period	365d 00h 00m	[1 hour - +inf]
Key Purpose	Client	
Certificate Store	Signing Record Store	
keyType	Certificate Parameter	
minKeySize	Certificate Parameter	

Transports

With **Transports**, certificate requester components can communicate and onboard with the Signing Service. This includes the mechanism by which clients authenticate to use the Signing Service, and how CSRs are associated with the relevant profile.



Transports	Signing Transport Folder	
foxTransport	Fox Signing Transport	
Enabled	● true	
Session Token Store	Session Token Store	
Temporary Code Expiration Period	00024h 00m 00s	[1 second - +inf]
Token Expiration Period	00001h 00m 00s	[1 second - +inf]
Expired Token Linger Period	00001h 00m 00s	[1 second - +inf]

Fox Signing Transport

This transport allows remote components to request a certificate from the Signing Service. To use this transport, the station on which the remote component is running will require an active Niagara network connection to the station that is hosting the Signing Service. Remote components must obtain a temporary session token for the purpose of onboarding, and each session token must be approved by an admin user who also designates a profile to service the request.

Local Signing Transport

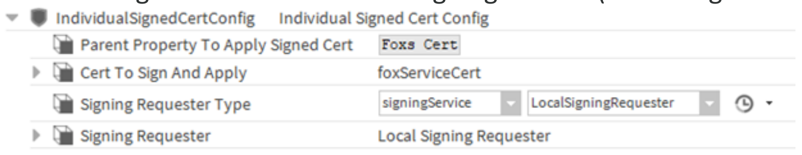
This transport was added in Niagara 4.14 and allows components to utilize a Signing Service that exists on the same station. No session token is required with this transport.

Signing Requester

A component on the remote station that will submit the CSR to the transport.

Individual Signed Cert Config

A component that is dropped from the Signing Service palette onto certain specific components, such as the Fox Service, Web Service, or an Additional Https Cert. This enables the parent component to make requests for an individual signed certificate from a Signing Service (as of Niagara 4.14).



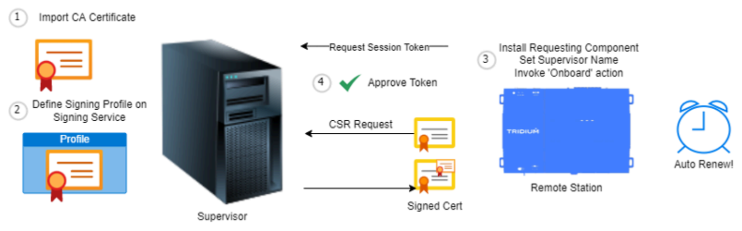
IndividualSignedCertConfig	Individual Signed Cert Config	
Parent Property To Apply Signed Cert	Foxs Cert	
Cert To Sign And Apply	foxServiceCert	
Signing Requester Type	signingService	LocalSigningRequester
Signing Requester	Local Signing Requester	

Combined Signed Cert Config

A component that is dropped from the Signing Service palette onto the station's Security Service. This enables the service to request a signed certificate that may then be shared between any combination of the Fox Service, Web Service, Platform or all three (as of Niagara 4.14).
Parent topic: Signing Service

Signing Service high level workflow

This example assumes the usage of the **Fox Signing Transport** component.



Since Niagara 4.14, it is also possible to onboard components that exist within the same station as the Signing Service. The workflow is the same except for the following key differences:

- It is not required to set the Supervisor name on the requesting component.
- It is not required for an administrator to approve the request. Instead, the user who invokes the **Onboard** action can select which Singing Profile should process the CSR. The user must have admin/ write permissions on the selected profile for the operation to succeed.

- **Configuring Signing Service**
  The following section describes the **Signing Service** high level workflow on a Supervisor (or non-Supervisor station that is designated as a station to which signing requests may be submitted).
- **Using Signing Service on remote station**
  The following section describes the **Signing Service** high level workflow on a remote station.
- **Using Signing Service on the same station**
  The following section describes the Signing Service high-level workflow when onboarding a component on the same station.

Parent topic: Signing Service

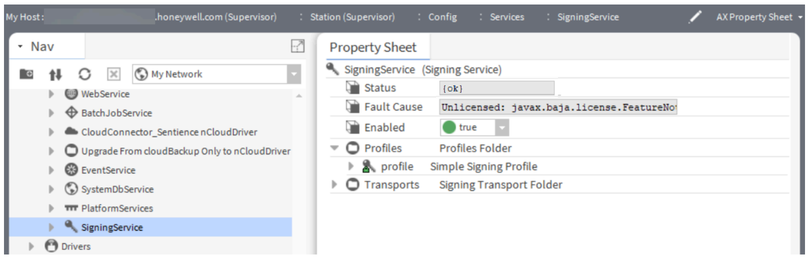
Configuring Signing Service

The following section describes the **Signing Service** high level workflow on a Supervisor (or non-Supervisor station that is designated as a station to which signing requests may be submitted).

- Niagara 4.13 and later module versions: signingService-rt, signingService-ux, and signingService-wb
- The "certSigningService" Tridium feature is included in your license.
- A valid CA certificate is a prerequisite for any signing operation with the Signing Service. You may choose to import an existing corporate CA certificate into the stations User Key Store (which requires both the public certificate and private key). This may be a self-signed corporate CA or an intermediate CA certificate signed by an external authority. Alternatively, you may use the Workbench Certificate Manager tool to generate a new CA. For more information about the Certificate Management tool, see "Platform Certificate Management (platCrypto-CertManagerService)" in the Niagara Platform Guide.

- If you are using the FOX transport to onboard components on remote Niagara stations, each remote station intending to use the Signing Service must have an active Niagara Network connection to the Supervisor station that hosts the service.

1. Import the CA certificate into the Supervisor via the Platform's **Certificate Management** view, or via the station at slot:/Services/PlatformServices/CertManagerService.
2. Add a **Signing Service** and a child **Signing Profile** to the **Services** folder. If you use the Signing Service in the **Typical Configuration** folder of the palette, it will already come with two profiles.


My Host : [redacted] .honeywell.com (Supervisor) : Station (Supervisor) : Config : Services : SigningService
AX Property Sheet
Nav
My Network
WebService
BatchJobService
CloudConnector_Sentience nCloudDriver
Upgrade From cloudBackup Only to nCloudDriver
EventService
SystemDbService
PlatformServices
SigningService
Drivers
Property Sheet

SigningService (Signing Service)

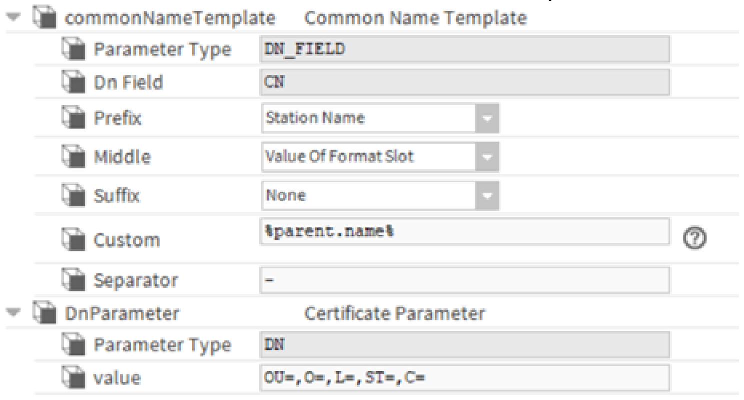
Status	{ok}
Fault Cause	Unlicensed: javax.baja.license.FeatureNo
Enabled	true
Profiles	Profiles Folder
profile	Simple Signing Profile
Transports	Signing Transport Folder

3. On the profile, select the alias of the imported CA, enter its unique password or global certificate password, and define any certificate parameters required.
   The fixed parameters allow certificate expiration and key purpose to be enforced. Other optional parameters in the palette allow enforcement of key size and key type, and certificate parameters such as the Distinguished Name fields.

   The **Typical Configuration** folder in the palette contains a good example on how to configure the Signing Service:

   - A clientProfile exists, which you can use for components that require a signing client SSL certificate for authentication purposes.

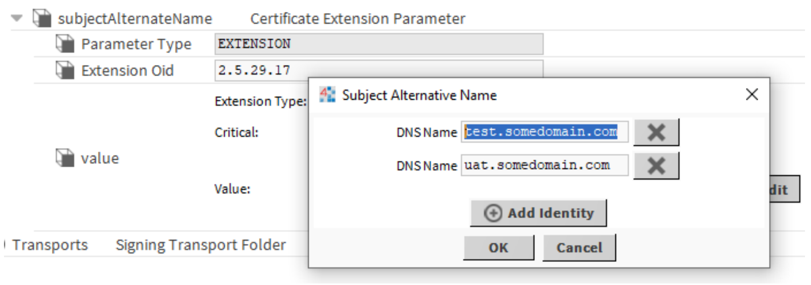
   - The **Common Name Template** parameter generates a dynamic Common Name (CN) value specific to the requesting component when the CSR is generated. You have several options to configure a CN value from multiple different sources, such as the station name, host name, or the value of a resolved user-defined format expression.



commonNameTemplate	Common Name Template
Parameter Type	DN_FIELD
Dn Field	CN
Prefix	Station Name
Middle	Value Of Format Slot
Suffix	None
Custom	%parent.name%
Separator	-
DnParameter	Certificate Parameter
Parameter Type	DN
value	OU=,O=,L=,ST=,C=

   - The DN Certificate Parameter allows you to further define the Distinguished Name values in a comma-separated list of pairs in the format: 'key=value'. You may choose to delete the **Common Name Template** parameter and add a hard-coded CN value within the DN parameter.

   The **Typical Configuration** folder also includes a serverProfile example for components that may supply a server SSL certificate. A **Common Name Template** is provided, along with a subjectAlternativeName (SAN) extension parameter. A SAN extension is commonly used to list the domain names / IP addresses for the host in the context of a server application.


subjectAlternateName Certificate Extension Parameter
Parameter Type EXTENSION
Extension Oid 2.5.29.17
Extension Type:
Critical:
value
Value:
Transports Signing Transport Folder

Subject Alternative Name
DNS Name test.somedomain.com
DNS Name uat.somedomain.com

You can edit this extension and add one to many different user-defined SAN values. In addition, your SAN parameters use one of the following Format token strings that will be substituted dynamically when the CSR is generated:

%commonName%	The value of the certificate's Common Name as supplied by another parameter in the request.
%hostnameOrIpAddress%	The host name of the device on which the requesting component's station is running, or the IP address if a non-localhost host name is not available. When falling back to the IP address, the SAN entry will be set to have an 'IP Address' Identity Type.
%hostname%	The host name of the device on which the requesting component's station is running.
0.0.0.0	The IP address of the host on which the requesting component's station is running. The SAN entry must have the 'IP Address' Identity Type.
%ipAddress%	The IP address of the host on which the requesting component's station is running. Works in conjunction with the 'DNS Name' Identity Type.

Other certificate parameters and extension types are supported and available in the palette.

Note:

Certificate Parameters may be dropped onto the requesting component, such as the **Individual Signing Cert Config** and the **Combined Signing Cert Config** components, not just the Signing Profile. If the same parameter type is added to both, the requester and the profile, the profile parameter value is the one that is applied. The exception to this rule is that different SAN extension values supplied are to be merged.

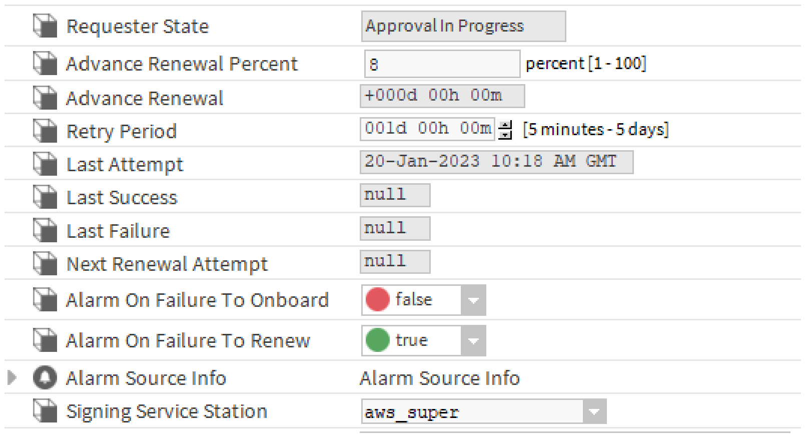
Parent topic: Signing Service high level workflow

Using Signing Service on remote station

The following section describes the **Signing Service** high level workflow on a remote station.

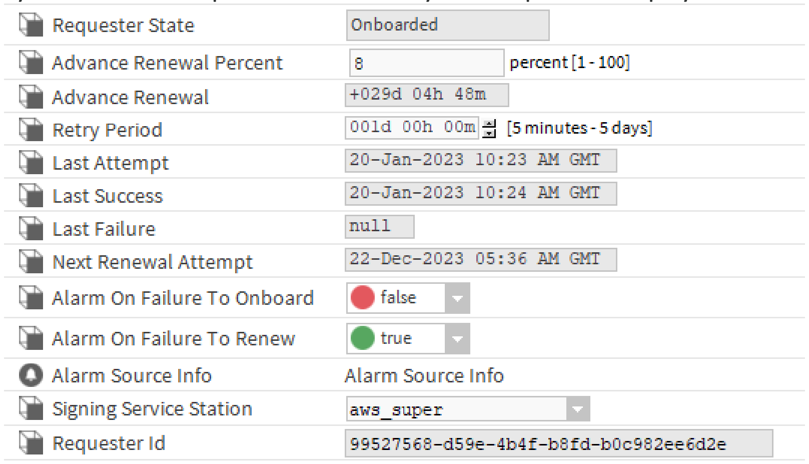
- The Niagara 4.13 and later module version: signingService-rt
- The Signing Service is pre-configured on at least one (other) remote station.
- Each remote station intending to use the **Signing Service** is required to have an active Niagara Network connection to the Supervisor station that hosts the service.

1. Install a requesting component on the station (see "*Installing a Signing Requester*").



Requester State	Approval In Progress	
Advance Renewal Percent	8	percent [1 - 100]
Advance Renewal	+000d 00h 00m	
Retry Period	001d 00h 00m	[5 minutes - 5 days]
Last Attempt	20-Jan-2023 10:18 AM GMT	
Last Success	null	
Last Failure	null	
Next Renewal Attempt	null	
Alarm On Failure To Onboard	🔴 false	
Alarm On Failure To Renew	🟢 true	
Alarm Source Info	Alarm Source Info	
Signing Service Station	aws_super	

2. Expand the Signing Requester configuration and select the name of the Supervisor that contains the service in the Signing Service station property.
3. You can now invoke the **Onboard** action and enter a comment to help the admin user on the Supervisor approve your request.
   If successful, the status will change to `Approval In Progress`. The **Signing Requester** will now communicate with the **Signing Service Transport** and generate a session token.
4. On the Supervisor, an admin user navigates to the **Session Token Store** for the **Fox Signing Transport** and inspects the request metadata and comment, then decides to approve or reject the request.
   Once an admin user of the Signing Service has approved your session token, the requesting component will generate and store a CSR locally, and submit this to the Signing Service. It will receive back a signed certificate, which is then stored in the main station's **Certificate Management > User Key Store**. The system will attempt to automatically renew prior to expiry.



Requester State	Onboarded	
Advance Renewal Percent	8	percent [1 - 100]
Advance Renewal	+029d 04h 48m	
Retry Period	001d 00h 00m	[5 minutes - 5 days]
Last Attempt	20-Jan-2023 10:23 AM GMT	
Last Success	20-Jan-2023 10:24 AM GMT	
Last Failure	null	
Next Renewal Attempt	22-Dec-2023 05:36 AM GMT	
Alarm On Failure To Onboard	🔴 false	
Alarm On Failure To Renew	🟢 true	
Alarm Source Info	Alarm Source Info	
Signing Service Station	aws_super	
Requester Id	99527568-d59e-4b4f-b8fd-b0c982ee6d2e	

Note: The exact steps may differ with future transport implementations. See specific component documentation for these.

Renewal will be automatically attempted prior to the certificate's expiration according to the Advance Renewal Percent value. This defaults to 8% so that an annual certificate will automatically attempt to renew approximately one month in advance of expiration. The existing certificate will now be replaced in the **User Key Store**. Approval is not required for renewal as the existing certificate is used as authentication with the service. However, if renewal fails before the existing certificate expires, you will need to manually repeat the **Onboard** action for the component. You can also manually invoke the **Renew** action to force an attempt.

An alarm will be generated on the local station for certificates that have failed a renewal attempt. The component will continue to schedule new renewal attempts in the case of a failure according to the Retry Period.

Some possibly causes for onboarding failure include:

- The request was rejected by the admin user on the Signing Service.
- The request was not approved in time.
- The CSR failed validation due to an incorrect value.
- A configuration error in the Signing Profile, for example, an incorrect CA password.
- Signing Service Station property was not populated.
- Communications failure with the Supervisor. Is the Niagara network connection functioning?
- A certificate alias conflict, whereby an active certificate with matching alias exists in the local key store during onboarding. In this instance, the certificate will be overwritten only if it has expired, and the password for the certificate matches. When using the **Individual Signing Cert Config** or **Combined Signing Cert Config** components, the existing certificate will be overwritten if the password matches, even if the current certificate has not expired.

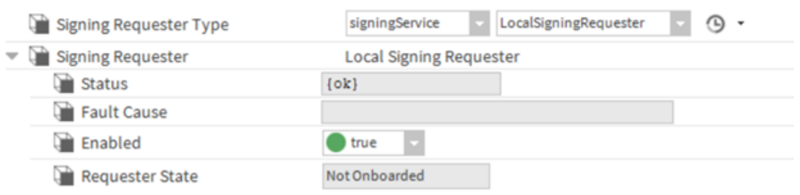
Parent topic: Signing Service high level workflow

Using Signing Service on the same station

The following section describes the Signing Service high-level workflow when onboarding a component on the same station.

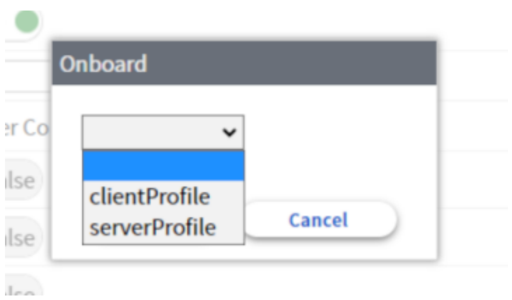
- Niagara module version: signingService-rt
- The Signing Service is pre-configured on the same station (also see "*Configuring the Signing Service*").

1. Install a Signing Requester component on the station (also see "*Installing a Signing Requester*").
2. From the Signing Requester Type drop-down menu, select LocalSigningRequester and expand **Signing Requester**.

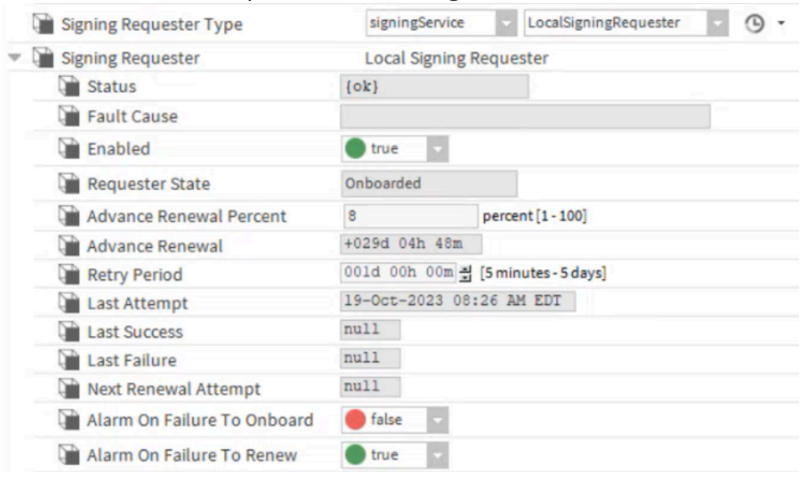


Signing Requester Type	signingService / LocalSigningRequester
Signing Requester	Local Signing Requester
Status	{ok}
Fault Cause	
Enabled	● true
Requester State	Not Onboarded

3. To register this component with the Signing Service, right-click on Individual or Combined Signed Cert Config and select **Actions > Onboard**
The **Onboard** window opens.
4. Select which Signing Profile should be used to process the generated CSR. Note that you need admin/ write permission on the selected profile for the operation to succeed.



If successful, the status will change to `Onboarded`, and renewal will be automatically attempted prior to the certificate's expiration according to the value in the Advance Renewal Percent property.



Signing Requester Type	signingService / LocalSigningRequester
Signing Requester	Local Signing Requester
Status	{ok}
Fault Cause	
Enabled	● true
Requester State	Onboarded
Advance Renewal Percent	8 percent [1 - 100]
Advance Renewal	+029d 04h 48m
Retry Period	001d 00h 00m [5 minutes - 5 days]
Last Attempt	19-Oct-2023 08:26 AM EDT
Last Success	null
Last Failure	null
Next Renewal Attempt	null
Alarm On Failure To Onboard	● false
Alarm On Failure To Renew	● true

Parent topic: Signing Service high level workflow

Installing a Signing Requester

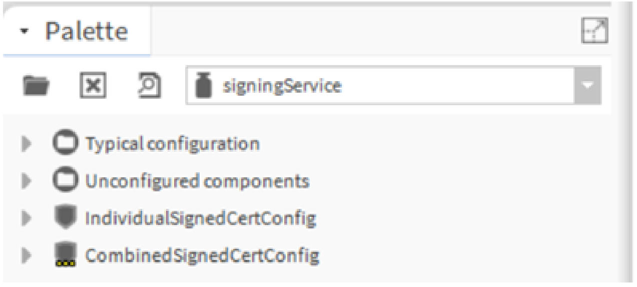
There are different ways to add the Signing Requester to station components.

Option 1

Add to the station a component that already has a certificate requester as an existing frozen property or dynamic property in the palette, such as the **Default MQTT Device**.

Option 2

From the palette, drag the **Individual Signed Cert Config** onto an existing supported component. These include the Fox Service, Web Service or the **Additional Https Certs** component.

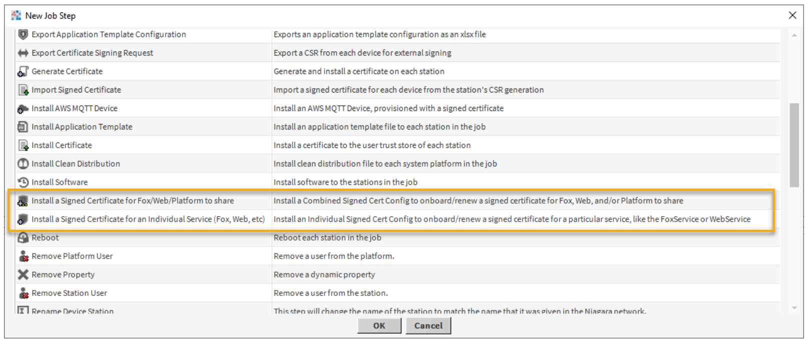


Option 3

From the palette, drag the **Combined Signed Cert Config** onto the station's Security Service. Select which of the core services to apply the signed certificate to.

Option 4

You can automatically install an **Individual Signed Cert Config** or **Combined Signed Cert Config** on several remote Niagara stations at once, using one of two provisioning steps. Onboarding to the Signing Service will take place automatically without the need of further action on the target station.


New Job Step

Export Application Template Configuration	Exports an application template configuration as an xlsx file
Export Certificate Signing Request	Export a CSR from each device for external signing
Generate Certificate	Generate and install a certificate on each station
Import Signed Certificate	Import a signed certificate for each device from the station's CSR generation
Install AWS MQTT Device	Install an AWS MQTT Device, provisioned with a signed certificate
Install Application Template	Install an application template file to each station in the job
Install Certificate	Install a certificate to the user trust store of each station
Install Clean Distribution	Install clean distribution file to each system platform in the job
Install Software	Install software to the stations in the job
Install a Signed Certificate for Fox/Web/Platform to share	Install a Combined Signed Cert Config to onboard/renew a signed certificate for Fox, Web, and/or Platform to share
Install a Signed Certificate for an Individual Service (Fox, Web, etc)	Install an Individual Signed Cert Config to onboard/renew a signed certificate for a particular service, like the FoxService or WebService
Reboot	Reboot each station in the job
Remove Platform User	Remove a user from the platform.
Remove Property	Remove a dynamic property
Remove Station User	Remove a user from the station.
Rename Device Station	This step will change the name of the station to match the name that it was given in the Niagara network.

- **Installing Signing Requester using provisioning job step**

Parent topic: Signing Service

Installing Signing Requester using provisioning job step

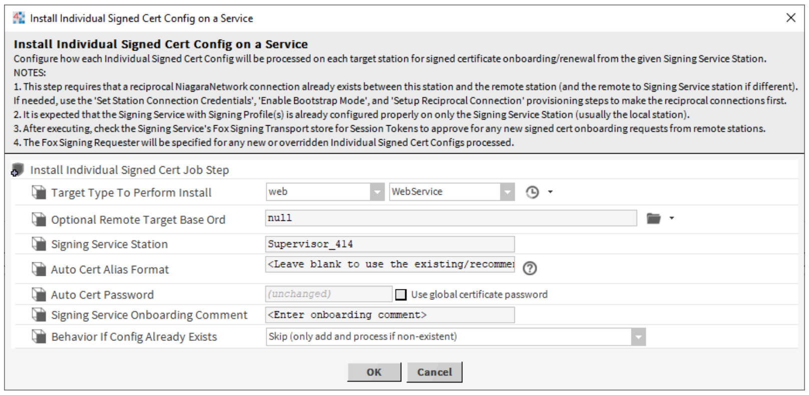
- Each target station is connected to a Niagara network.
- A reciprocal Niagara network connection already exists between the target station and the Signing Service station, and also the station running the provisioning job step if it is a different station. If you have not yet established these reciprocal connections, there are other existing provisioning job steps that can help you with the setup.
- You have installed and configured the Signing Service and profile(s) with a CA certificate.

To help bootstrap the Niagara network connections, some of the following provisioning steps may be run prior to or as part of the same provisioning job step:

1. Set the credentials for the station connection. For more information, see “*Install Certificate*” in the Niagara Provisioning Guide.
2. Enable bootstrap mode. It allows a temporary Supervisor-to-device connection using a certificate exemption for the device’s self-signed certificate. For more information, see “*Enable Bootstrap Mode*” in the Niagara Provisioning Guide.
3. Install certificate, which can be used to copy a certificate from the User Trust Store of the Supervisor to each station.
   Copying the CA or Intermediate CA that signed the Supervisor’s Fox Server certificate means that no approval of the connection is required on the remote controller. If the CA has been signed by a well-known external CA that pre-exists in the System Trust Store, this step will not be required. Ensure the Common Name of the Supervisor’s Fox Certificate matches the IP/hostname used to connect back.
   For more information, see “*Install Certificate*” in the Niagara Provisioning Guide.
4. Set up reciprocal connection, which establishes the Niagara network connection from the remote device back to the Supervisor. For more information, see “*Set up Reciprocal Connection*” in the Niagara Provisioning Guide.
   A single provisioning procedure is not explained here, as this may vary depending on what assets already exist on the remote station.

Install Individual Signed Cert Job Step:

5. Select from the Target Type To Perform Install drop-down list the individual component type that you want to target on the remote station.



Install Individual Signed Cert Config on a Service

Install Individual Signed Cert Config on a Service
Configure how each Individual Signed Cert Config will be processed on each target station for signed certificate onboarding/renewal from the given Signing Service Station.
NOTES:
1. This step requires that a reciprocal NiagaraNetwork connection already exists between this station and the remote station (and the remote to Signing Service station if different).
If needed, use the 'Set Station Connection Credentials', 'Enable Bootstrap Mode', and 'Setup Reciprocal Connection' provisioning steps to make the reciprocal connections first.
2. It is expected that the Signing Service with Signing Profile(s) is already configured properly on only the Signing Service Station (usually the local station).
3. After executing, check the Signing Service's Fox Signing Transport store for Session Tokens to approve for any new signed cert onboarding requests from remote stations.
4. The Fox Signing Requester will be specified for any new or overridden Individual Signed Cert Configs processed.

Install Individual Signed Cert Job Step

Target Type To Perform Install	web / WebService
Optional Remote Target Base Ord	null
Signing Service Station	Supervisor_414
Auto Cert Alias Format	<Leave blank to use the existing/recomme
Auto Cert Password	(unchanged) ☐ Use global certificate password
Signing Service Onboarding Comment	<Enter onboarding comment>
Behavior If Config Already Exists	Skip (only add and process if non-existent)

- To reduce the search scope for target components, you can use the Optional Remote Target Base Ord. Null will result in an entire station search.

- Optionally, in Auto Cert Alias Format, you can override the alias for generated certificates or leave blank to accept the default.

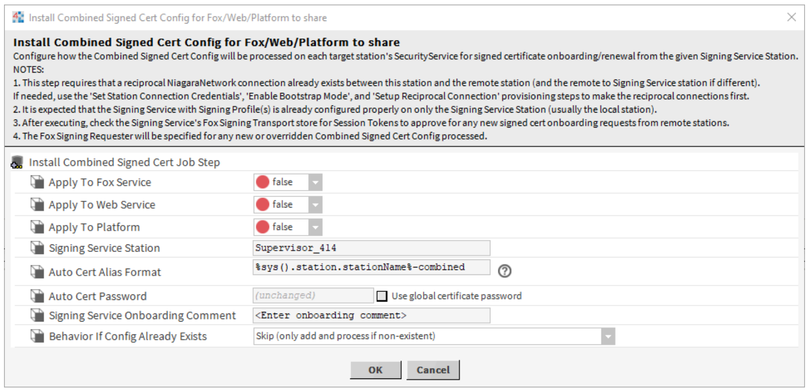
- For the Auto Cert Password property, it is recommended to enter a password to secure the signing certificates.

- The Signing Service Onboarding Comment will be presented to the admin user who approves the process in the Signing Service's Fox Transport. Consequently, enter a comment that identifies the reason as to why a signed certificate is required. If you leave the default value, an auto-generated comment with details about the requesting component will be used.

- The Behavior If Config Already Exists property allows you to decide how the step should behave if the config already exists on a given target component: You can choose if you want to skip, overwrite, just invoke the onboard, or renew action on the existing config.

Install Combined Signed Cert Job Step

6. Select which of the three core services you want to associate with the signed certificate by setting them to true. This step is very similar to **Install Individual Signed Cert Job Step** with the exception that the default alias here is pre-populated with a suggested format.



Install Combined Signed Cert Config for Fox/Web/Platform to share

Install Combined Signed Cert Config for Fox/Web/Platform to share
Configure how the Combined Signed Cert Config will be processed on each target station's SecurityService for signed certificate onboarding/renewal from the given Signing Service Station.
NOTES:
1. This step requires that a reciprocal NiagaraNetwork connection already exists between this station and the remote station (and the remote to Signing Service station if different).
If needed, use the 'Set Station Connection Credentials', 'Enable Bootstrap Mode', and 'Setup Reciprocal Connection' provisioning steps to make the reciprocal connections first.
2. It is expected that the Signing Service with Signing Profile(s) is already configured properly on only the Signing Service Station (usually the local station).
3. After executing, check the Signing Service's Fox Signing Transport store for Session Tokens to approve for any new signed cert onboarding requests from remote stations.
4. The Fox Signing Requester will be specified for any new or overridden Combined Signed Cert Config processed.

Install Combined Signed Cert Job Step

Apply To Fox Service	🔴 false
Apply To Web Service	🔴 false
Apply To Platform	🔴 false
Signing Service Station	Supervisor_414
Auto Cert Alias Format	%sys().station.stationName%-combined
Auto Cert Password	(unchanged) ☐ Use global certificate password
Signing Service Onboarding Comment	<Enter onboarding comment>
Behavior If Config Already Exists	Skip (only add and process if non-existent)

Parent topic: Installing a Signing Requester

Components, views and windows

The user interface includes components, views and windows, which provide the means for communicating with the system.

The Help topics include context sensitive information about each component and view, as well as information about individual windows.

- **Components**
  Components include services, folders and other model building blocks associated with a module. You may drag them to a property or wire sheet from a palette.
- **Plugins (views)**
  Plugins provide views of components and can be accessed in many ways. For example, double-click a component in the Nav tree to see its default view. In addition, you can right-click on a component and select from its **Views** menu.

Components

Components include services, folders and other model building blocks associated with a module. You may drag them to a property or wire sheet from a palette.

Descriptions included in the following topics appear as context-sensitive help topics when accessed by:

- Right-clicking on the object and selecting **Views > Guide Help**
- Clicking **Help > Guide On Target**
- **Signing Service (signingService-SigningService)**
  The **Signing Service** is intended primarily for the use on a Niagara Supervisor. As of Niagara 4.13, it allows components within remote stations to make secure Certificate Signing Requests (CSR) to obtain signed X509 certificates, then additional requests to renew these certificates before they expire. The **Signing Service** fulfills these requests by returning certificates signed by a designated Certificate Authority (CA).
- **Simple Signing Profile (signingService-SimpleSigningProfile)**
  A signing service profile is where you define the alias and password of the CA certificate that will be used to sign all Certificate Signing Requests (CSRs) associated with that profile. It is also where you can define values for various certificate fields that are applied to the signed certificate, and to validate those fields within the CSR supplied by the remote station.
- **Signing Record Store (signingService-SigningRecordStore)**
  The certificate **Signing Record Store** holds and manages the records of outstanding and fulfilled CSR requests for the parent signing profile.
- **Certificate Signing Record (signingService-CertificateSigningRecord)**
  This component represents an individual signing record associated with a requesting component. It encapsulates the status of the signing request and the status of the certificate.
- **Fox Signing Transport (signingService-FoxSigningTransport)**
  The **Fox Signing Transport** is a signing transport that allows remote stations to utilize the Signing Service via their Fox Niagara network connection. A dedicated Fox Signing message channel provides a secure and trusted communication mechanism between the Signing Service and the remote requesting component for exchange of CSR and signed certificates.
- **Local Signing Transport (signingService-LocalSigningTransport)**
  The **Local Signing Transport** is a signing transport that allows components to utilize a Signing Service that exists in the same station.
- **Session Token (signingService-SessionToken)**
  Each **Session Token** represents a requesting component which has attempted to onboard with the Signing Service to submit a CSR.
- **Fox Signing Requester (signingService-FoxSigningRequester)**

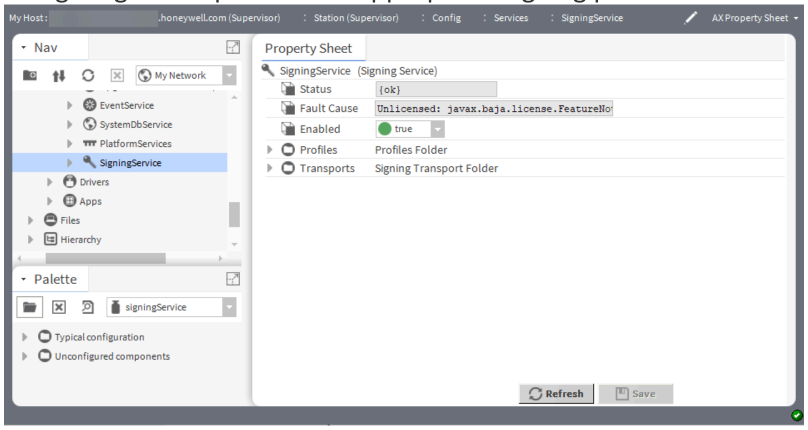
This component is a client to the **Fox Signing Transport** and provides a parent component with the ability to make requests to onboard with the Signing Service and to renew certificates.
- **Local Signing Requester (signingService-LocalSigningRequester)**
  This component is a client to the Local Signing Transport and provides a parent component with the ability to make requests to onboard/renew with a Signing Service running in the same station.
- **Individual Signed Cert Config (signingService-IndividualSignedCertConfig)**
  The **Individual Signed Cert Config** allows you to define the alias and password for the certificate that you use for the signing request.
- **Common Name Template (platCrypto-CommonNameTemplate)**
  The **Common Name Template** component is a type of Certificate Parameter that will generate a dynamic common name for the signed certificate based upon configurable settings. The main benefit of this is the ability to generate a potentially unique common name for each component.
- **Certificate Extension Parameter (platCrypto-CertificateExtensionParameter)**
  The **Certificate Extension Parameter** component allows you to configure a Certificate Extension that will be applied to the CSR to be submitted for signing.
- **Combined Signed Cert Config (signingService-CombinedSignedCertConfig)**
  The **Combined Signed Cert Config** allows you to define the alias and password for the certificate that you use for the signing request.

Parent topic: Components, views and windows

Signing Service (signingService-SigningService)

The **Signing Service** is intended primarily for the use on a Niagara Supervisor. As of Niagara 4.13, it allows components within remote stations to make secure Certificate Signing Requests (CSR) to obtain signed X509 certificates, then additional requests to renew these certificates before they expire. The **Signing Service** fulfills these requests by returning certificates signed by a designated Certificate Authority (CA).

This component is used to fulfill the requests of queuing incoming CSR requests from signing transports, and delegating the request to the appropriate signing profile.

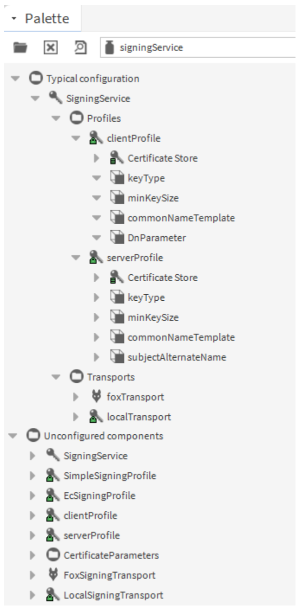


SigningService (Signing Service)

Status	{ok}
Fault Cause	Unlicensed: javax.baja.license.FeatureNo
Enabled	● true
Profiles	Profiles Folder
Transports	Signing Transport Folder

To use the **Signing Service** component, drag it from the signingService palette to the **Config > Services** folder in the Nav tree.

You also find an example of a **Signing Service** pre-configured with an example signing profile and transport in the **Typical configuration** folder. There is also a plain unconfigured service in the **Unconfigured components** folder.



Palette

signingService

- Typical configuration
  - SigningService
    - Profiles
      - clientProfile
        - Certificate Store
        - keyType
        - minKeySize
        - commonNameTemplate
        - DnParameter
      - serverProfile
        - Certificate Store
        - keyType
        - minKeySize
        - commonNameTemplate
        - subjectAlternateName
    - Transports
      - foxTransport
      - localTransport
- Unconfigured components
  - SigningService
  - SimpleSigningProfile
  - EcSigningProfile
  - clientProfile
  - serverProfile
  - CertificateParameters
  - FoxSigningTransport
  - LocalSigningTransport

To access the properties, expand **Config > Services** and click **SigningService**

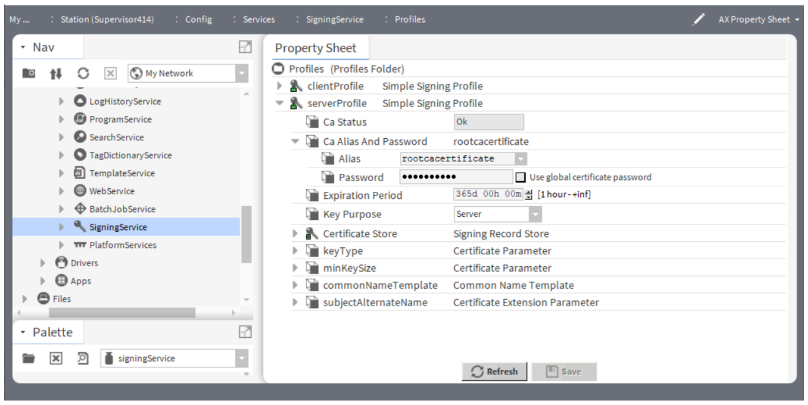
Property	Value	Description
Enabled	true or false (defaults to true)	When true, all **Signing Service** functionality is enabled. While false, all submitted CSR will fail to process, and any queued CSR will be rejected.
Profiles	folder	Holds signing profile instances. See "*Signing Service High Level Types*" for more details.
Transports	folder	Holds signing profile instances. See "*Signing Service High Level Types*" for more details.

Parent topic: Components

Simple Signing Profile (signingService-SimpleSigningProfile)

A signing service profile is where you define the alias and password of the CA certificate that will be used to sign all Certificate Signing Requests (CSRs) associated with that profile. It is also where you can define values for various certificate fields that are applied to the signed certificate, and to validate those fields within the CSR supplied by the remote station.

Each profile also holds a **Certificate Store** where you can view the records of a CSR associated with that profile, including the signed certificate.



Profiles (Profiles Folder)	
clientProfile	Simple Signing Profile
serverProfile	Simple Signing Profile
Ca Status	Ok
Ca Alias And Password	rootcacertificate
Alias	rootcacertificate
Password	••••••••••• ☐ Use global certificate password
Expiration Period	365d 00h 00m [1 hour - +inf]
Key Purpose	Server
Certificate Store	Signing Record Store
keyType	Certificate Parameter
minKeySize	Certificate Parameter
commonNameTemplate	Common Name Template
subjectAlternateName	Certificate Extension Parameter

The **Typical configuration** folder contains signing profiles predefined for signing RSA certificates with a minimum key size and various other useful default parameters. There is also a plain unconfigured profile without any parameters in the **Unconfigured components** folder.

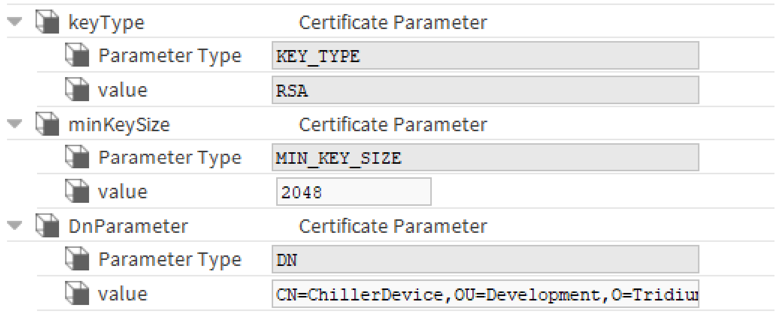
For a signing profile to fulfill requests, the following prerequisites must be met:

- The CA certificate has been imported into the station or platform certificate manager.
- The alias and password of the CA have been set in the Signing Profile’s Ca Alias And Password property. A password is mandatory.

Note: Do not select the Use global certificate password checkbox as this option is not valid for the CA certificate.

You may set fixed certificate values such as the expiration period of the signed certificates and the key purpose. These values override those set in the incoming CSR from the requesting component.

You have also the option to drop one of many certificate parameter objects underneath the signing profile. These allow you to define default values for any fields that may be unspecified in the incoming CSR. A validation error will be thrown if the incoming CSR defines a different value, causing the signing request to be rejected. Parameters include key type and minimum key size, and certificate parameters such as the Distinguished Name fields.



keyType	Certificate Parameter
Parameter Type	KEY_TYPE
value	RSA
minKeySize	Certificate Parameter
Parameter Type	MIN_KEY_SIZE
value	2048
DnParameter	Certificate Parameter
Parameter Type	DN
value	CN=ChillerDevice,OU=Development,O=Tridiu

In the **Typical configuration > SigningService** folder, the available profiles (**clientProfile** and **serverProfile**) are already preconfigured when adding the Signing Service. They come with values that are typically needed for two types of certificates commonly signed, that is, the client certificate and the server certificate. The **serverProfile**, for example, is used by the Fox Service, Web Service, and the Platform. Server certificates are issued by servers whose clients access them to validate the server. The **clientProfile** may be useful in cases in which a client certificate is needed to allow a server to validate an accessing client.

Property	Value	Description
Ca Status	read-only	Displays the current status of the Ca Alias And Password property configuration. The following states can be indicated: • Ok: the configured Ca Alias And Password property is valid for an existing CA certificate in the key store. • Bad Key: default value once a new (unconfigured) signing profile is added to a running station. This state means that the Ca Alias And Password property is currently configured with the `default` alias specified, or the alias field is empty, or the alias is specified to an existing certificate, but the certificate is not a proper CA. • Missing Key: the Ca Alias And Password property is currently configured with an alias that does not exist in the key store. • Bad Password: the Ca Alias And Password property is currently configured with a valid alias, but the password is incorrect for the selected CA certificate.
Ca Alias And Password	additional mandatory properties	Specifies the alias and password of the CA certificate that will be used to sign incoming CSR for this profile. Note: Do not select the Use global certificate password checkbox as this option is not valid for a CA certificate.
Expiration Period	days, hours, minutes	Defines the expiration period that will be applied to the signed certificate, overriding any specified in the CSR. The period is applied to the time at which the CSR is processed by the service.
Key Purpose	drop-down menu	Specifies the purpose certificate extensions that will be applied to the signed certificate, overriding any specified in the CSR. Values are Client, Server, Cert Authority and Code Signing.
Certificate Store	additional properties	Holds the records of CSR associated with that profile, including the signed certificate.
commonNameTemplate	additional properties (by default added to	For more information, see "*Common Name*

Property	Value	Description
	clientProfile and **serverProfile**)	*Template (platCrypto-CommonNameTemplate*" in the Niagara Station Security Guide.
subjectAlternateName	additional properties(by default added to **serverProfile**)	For more information, see "*Certificate Extension Parameter (platCrypto-CertificateExtensionParameter)*" in the Niagara Station Security Guide.

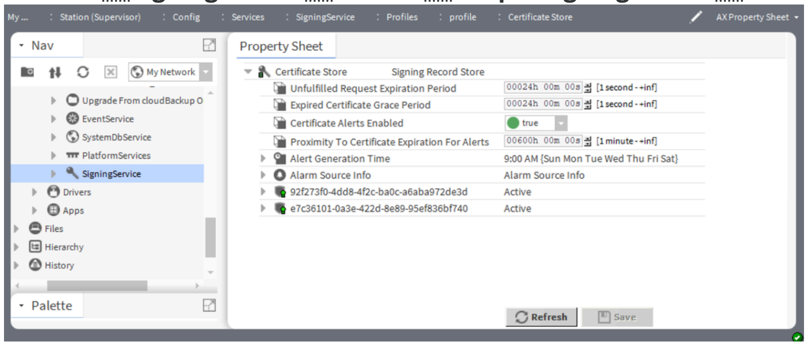
Parent topic: Components

Signing Record Store (signingService-SigningRecordStore)

The certificate **Signing Record Store** holds and manages the records of outstanding and fulfilled CSR requests for the parent signing profile.

When CSRs are submitted and processed, they will appear underneath the store as **Certificate Signing Record** instances specific to the requester Id of the requesting component.

The **Signing Record Store** is available in the signingService palette. To access the properties, expand **Config > Services > SigningService > Profiles > Simple Signing Profile > Certficate Store**.



```
Certificate Store                              Signing Record Store
  Unfulfilled Request Expiration Period        00024h 00m 00s [1 second - +inf]
  Expired Certificate Grace Period             00024h 00m 00s [1 second - +inf]
  Certificate Alerts Enabled                   true
  Proximity To Certificate Expiration For Alerts 00600h 00m 00s [1 minute - +inf]
  Alert Generation Time                        9:00 AM {Sun Mon Tue Wed Thu Fri Sat}
  Alarm Source Info                            Alarm Source Info
  92f273f0-4dd8-4f2c-ba0c-a6aba972de3d         Active
  e7c36101-0a3e-422d-8e89-95ef836bf740         Active
```

Property	Value	Description
Unfulfilled Request Expiration Period	hours, minutes, seconds	Defines the period after which a `Pending` record will be deleted.
Expired Certificate Grace Period	hours, minutes, seconds	Defines the period after which an `Expired` certificate will continue to service renewal requests.
Certificate Alerts Enabled	true or false	Set to true enables alerts for certificates that are due or have already expired.
Proximity To Certificate Expiration For Alerts	hours, minutes, seconds	Defines the period in which alerts shall be generated for certificates that are due to expire.
Alert Generation Time	additional properties	Contains time trigger settings for timings of alert checks.
Alarm Source Info	additional properties	Contains properties for configuring certificate alerts.

Actions

Update Record States: Forces the update of the status of each record in the store.

Generate Alerts: Forces an instant generation of new certificate alerts.

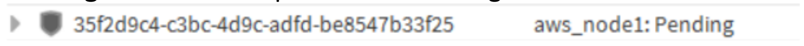
Requester certificate rules

The rules governing requester certificates are as follows:

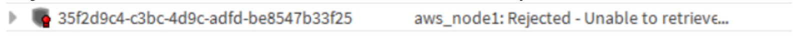
- The requester Id is unique to each requesting component. There may be several on the same station, or from different stations.
- Each requester can be associated with only a single signing record.
- The certificate record is `Pending` while the CSR is queued until the request is processed. Then, it becomes `Rejected` or `Active` as the certificate is signed and stored.
- Any pending request that is unfulfilled for the Unfulfilled Request Expiration Period is automatically removed. The requester must **Onboard** to the signing service once again.
- An active certificate record is required for the requesting component to renew its current certificate. This should be attempted automatically in advance of expiry but can also be requested manually.
- The certificate record will continue to service renewal attempts up to the expiry time of the certificate, plus the Expired Certificate Grace Period.
- After the certificate expiry plus grace period has passed, the expired certificate can no longer service renewal requests and is automatically removed. The requester must **Onboard** to the signing service once again.

The status for each record in the store is indicated by the icon and status text. The possible states are:

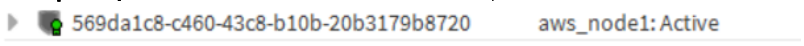
- **Pending** — The CSR is queued and waiting to be fulfilled.

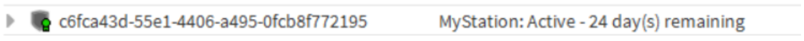

35f2d9c4-c3bc-4d9c-adfd-be8547b33f25 aws_node1: Pending

- **Rejected** — The CSR failed to be successfully fulfilled. Check the records message property for cause.

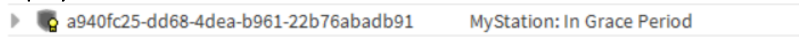

35f2d9c4-c3bc-4d9c-adfd-be8547b33f25 aws_node1: Rejected - Unable to retrieve...

- **Complete/Active** — The CSR was fulfilled, and the certificate is valid.


569da1c8-c460-43c8-b10b-20b3179b8720 aws_node1: Active


c6fca43d-55e1-4406-a495-0fcb8f772195 MyStation: Active - 24 day(s) remaining

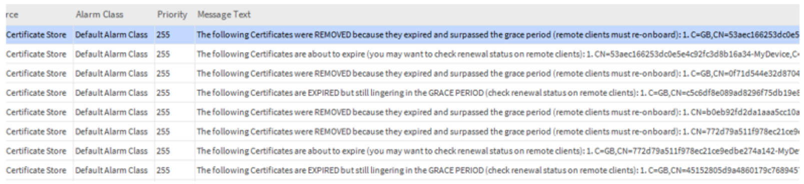
- **Complete/In Grace Period** — The CSR was previously fulfilled, the certificate has passed its expiry period, however is in the grace period, so it will continue to be used to service renewal requests until the Expired Certificate Grace Period has passed. The grace period is applied from the time of certificate expiry.


a940fc25-dd68-4dea-b961-22b76abadb91 MyStation: In Grace Period

The **Certificate Store** also has the ability to generate alerts for various certificate states. Certificates are grouped together into single alerts for each of the following states. They are grouped with those that have also entered that state since the last alert check:

- Those near their expiration; the proximity is configurable.

- Those expired but still in the grace period; this may indicate that auto-renew on the requesting component is failing.
- Those that have expired and been removed since the last alert check.



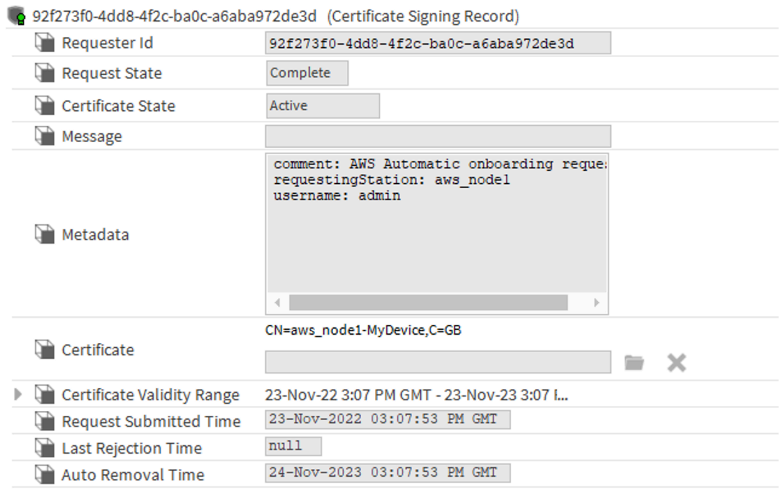
rce	Alarm Class	Priority	Message Text
Certificate Store	Default Alarm Class	255	The following Certificates were REMOVED because they expired and surpassed the grace period (remote clients must re-onboard): 1. C=GB,CN=53aec166253dc0e5
Certificate Store	Default Alarm Class	255	The following Certificates are about to expire (you may want to check renewal status on remote clients): 1. CN=53aec166253dc0e5e4c92fc3d8b16a34-MyDevice,C
Certificate Store	Default Alarm Class	255	The following Certificates were REMOVED because they expired and surpassed the grace period (remote clients must re-onboard): 1. C=GB,CN=0f71d544e32d8704
Certificate Store	Default Alarm Class	255	The following Certificates are EXPIRED but still lingering in the GRACE PERIOD (check renewal status on remote clients): 1. C=GB,CN=c5c6df8e089ad8296f75db19e
Certificate Store	Default Alarm Class	255	The following Certificates were REMOVED because they expired and surpassed the grace period (remote clients must re-onboard): 1. CN=b0eb92fd2da1aaa5cc10a
Certificate Store	Default Alarm Class	255	The following Certificates were REMOVED because they expired and surpassed the grace period (remote clients must re-onboard): 1. CN=772d79a511f978ec21ce9
Certificate Store	Default Alarm Class	255	The following Certificates are about to expire (you may want to check renewal status on remote clients): 1. C=GB,CN=772d79a511f978ec21ce9edbe274a142-MyDe
Certificate Store	Default Alarm Class	255	The following Certificates are EXPIRED but still lingering in the GRACE PERIOD (check renewal status on remote clients): 1. C=GB,CN=45152805d9a4860179c768945

Parent topic: Components

Certificate Signing Record (signingService-CertificateSigningRecord)

This component represents an individual signing record associated with a requesting component. It encapsulates the status of the signing request and the status of the certificate.

Each record displays the certificate itself, the metadata captured during onboarding to the Signing Service for identification, and various date/times key to the certificate's lifecycle.



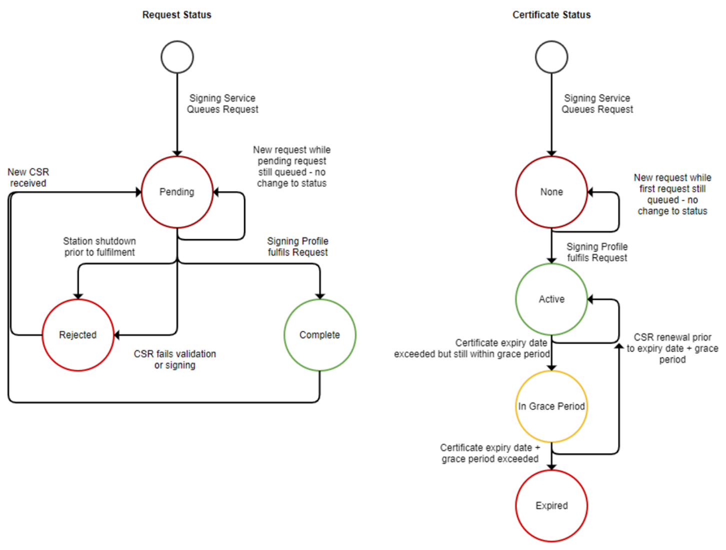
92f273f0-4dd8-4f2c-ba0c-a6aba972de3d (Certificate Signing Record)

Requester Id	92f273f0-4dd8-4f2c-ba0c-a6aba972de3d
Request State	Complete
Certificate State	Active
Message	
Metadata	comment: AWS Automatic onboarding reque requestingStation: aws_node1 username: admin
Certificate	CN=aws_node1-MyDevice,C=GB
Certificate Validity Range	23-Nov-22 3:07 PM GMT - 23-Nov-23 3:07 I...
Request Submitted Time	23-Nov-2022 03:07:53 PM GMT
Last Rejection Time	null
Auto Removal Time	24-Nov-2023 03:07:53 PM GMT

Property	Value	Description
Requester Id	read-only	Displays a unique Id that identifies the requesting component.
Request State	read-only	Displays the status of the request (`Pending`, `Rejected`, `Complete`)
Certificate State	read-only	Displays the status of the certificate (`None`, `Active`, `In Grace Period`, `Expired`)
Message	read-only	Displays message to identify reason for rejection or failure to onboard.
Metadata	read-only	Displays extra details regarding the

Property	Value	Description
		requester from the onboarding process, including station name and name of the user who submitted the request.
Certificate	read-only	Displays the certificate distinguished name.
Certificate Validity Range	read-only	Specifies the not-before and not-after times for the signed certificate.
Request Submitted Time	read-only	Specifies the time at which the CSR was submitted to the signing service.
Last Rejection Time	read-only	Specifies the time at which the CSR last failed to be fulfilled.
Auto Removal Time	read-only	Specifies the time at which the record will automatically be removed.

The status workflow is as follows:

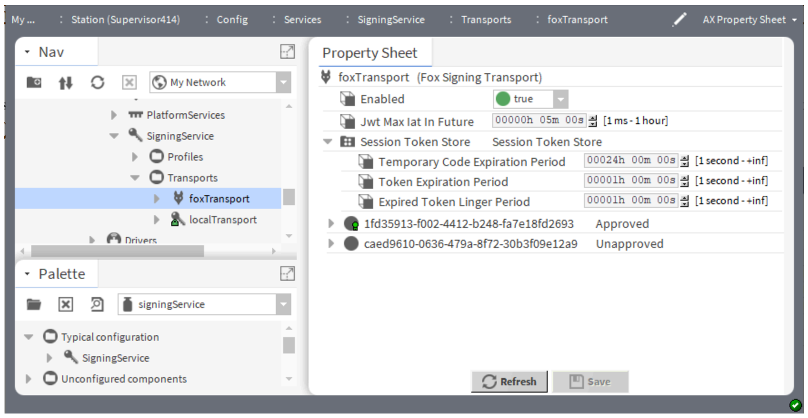

Request Status
Signing Service Queues Request
New request while pending request still queued - no change to status
New CSR received
Pending
Station shutdown prior to fulfilment
Signing Profile fulfils Request
Rejected
Complete
CSR fails validation or signing
Certificate Status
Signing Service Queues Request
None
New request while first request still queued - no change to status
Signing Profile fulfils Request
Active
Certificate expiry date exceeded but still within grace period
CSR renewal prior to expiry date + grace period
In Grace Period
Certificate expiry date + grace period exceeded
Expired

Parent topic: Components

Fox Signing Transport (signingService-FoxSigningTransport)

The **Fox Signing Transport** is a signing transport that allows remote stations to utilize the Signing Service via their Fox Niagara network connection. A dedicated Fox Signing message channel provides a secure and trusted communication mechanism between the Signing Service and the remote requesting component for exchange of CSR and signed certificates.

Each signing transport is responsible for onboarding remote components into the service. The **Fox Signing Transport** requires the remote component to request a temporary session token, which must be approved by an admin user before the component can submit a CSR to the service. These tokens are stored and approved in the **Session Token Store**. Session tokens are not required for certificate renewal, only for initial **Onboarding** with the service.



```
foxTransport  (Fox Signing Transport)
  Enabled                          true
  Jwt Max Iat In Future            00000h 05m 00s  [1 ms - 1 hour]
  Session Token Store              Session Token Store
    Temporary Code Expiration Period   00024h 00m 00s  [1 second - +inf]
    Token Expiration Period            00001h 00m 00s  [1 second - +inf]
    Expired Token Linger Period        00001h 00m 00s  [1 second - +inf]
  1fd35913-f002-4412-b248-fa7e18fd2693   Approved
  caed9610-0636-479a-8f72-30b3f09e12a9   Unapproved
```

The **Fox Signing Transport** component is available in the signingService palette. To access the properties, expand **Config > Services > SigningService > Transports > foxTransport (Fox Signing Transport)**.

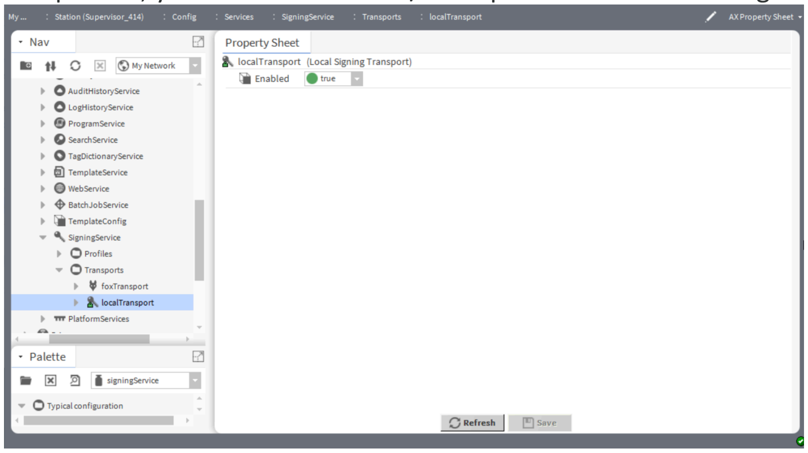
Property	Value	Description
Enabled	true or false (defaults to true)	Enables the Fox signing message channel communications.
Jwt Max Iat In Future	hour, minute, second (defaults to 5 minutes)	As of Niagara 4.14, specifies the maximum clock skew time. When the **Fox Signing Transport** is added to a running station, this property will become visible. During an onboarding or renewal process of a signed certificate for a remote station, if the system clocks between the platforms are not synchronized by more than this defined time window, the onboarding/renewal attempt may fail and display a similar fault cause: `GenerateCertificateAndSubmitCsr failed due to The JWT cannot be validated for the given requester ID (check configuration and/or re-onboard): XXX` You can configure this property to allow more (or less) skew time to approve the processing of a request from a remote station to onboard/renew a certificate. Note: For security reasons, keep this value as small as possible, but still allow enough skew time for the possibility of system clocks in the Niagara system being out of sync. It is also strongly recommended to use NTP (Network Time Protocol) in Niagara stations to keep the clocks in sync.
Session Token Store	additional properties	Contains session tokens, which permit a requesting component to onboard with the Signing Service.

Parent topic: Components

Local Signing Transport (signingService-LocalSigningTransport)

The **Local Signing Transport** is a signing transport that allows components to utilize a Signing Service that exists in the same station.

Session Tokens and approval are not required with this transport, however if you onboard the requesting component, you must have admin/write permissions on the target Signing Profile.



localTransport (Local Signing Transport)

Enabled	true

The **Local Signing Transport** component is available in the signingService palette. To access the properties, expand **Config > Services > SigningService > Transports > localTransport (Local Signing Transport)**.

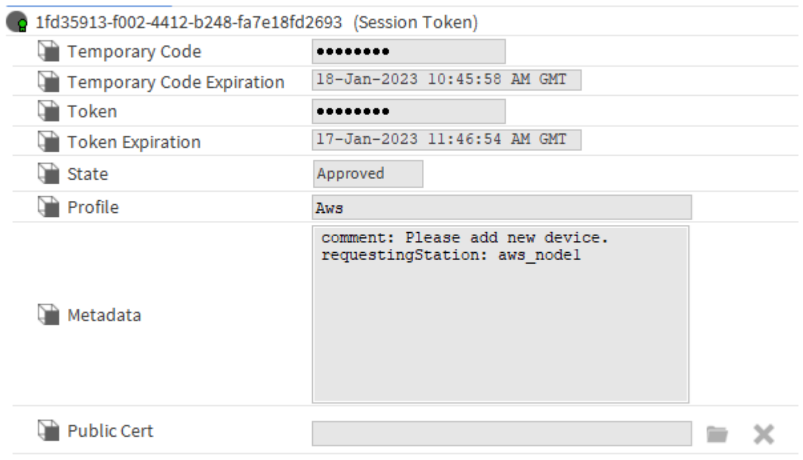
Property	Value	Description
Enabled	true or false (defaults to true)	Enables local signing from components that exist on the same station.

Parent topic: Components

Session Token (signingService-SessionToken)

Each **Session Token** represents a requesting component which has attempted to onboard with the Signing Service to submit a CSR.

The name of the token is the unique requested Id. Once the CSR has been fulfilled, the token's certificate property will be populated. Session tokens will be automatically removed some time after the CSR has been fulfilled.



1fd35913-f002-4412-b248-fa7e18fd2693 (Session Token)

Temporary Code	••••••••
Temporary Code Expiration	18-Jan-2023 10:45:58 AM GMT
Token	••••••••
Token Expiration	17-Jan-2023 11:46:54 AM GMT
State	Approved
Profile	Aws
Metadata	comment: Please add new device. requestingStation: aws_node1
Public Cert	

Property	Value	Description
Temporary Code	read-only	Displays the secret temporary code value. This enables the requesting component to check on the status of its token.
Temporary Code Expiration	read-only	Displays the date and time at which the code expires.
Token	read-only	Displays the secret token value which enables the requesting component to submit a CSR.
Token Expiration	read-only	Displays the date and time at which the token expires.
State	read-only	Displays the status of the token: `Unapproved/Approved/Rejected`
Profile	read-only	Displays the name of the profile that was associated with the approved token.
Metadata	read-only	Lists metadata values submitted by the requesting component, which should help inform the admin user on the decision to approve or reject the token.
Public Cert	read-only	Displays the distinguished name of the certificate. Appears when the CSR has been fulfilled.

Actions

Approve: Approves the selected token to allow the requesting component to use the signing service.

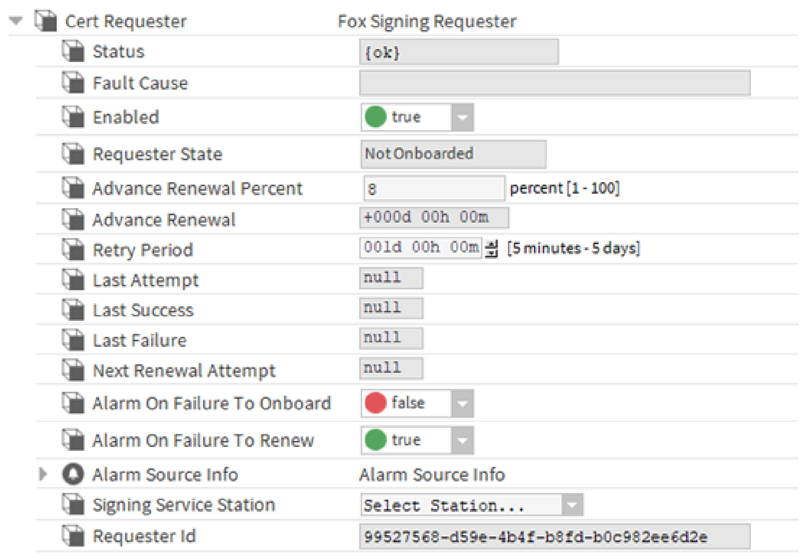
Reject: Rejects the selected token. The requesting component will not submit CSR.

Parent topic: Components

Fox Signing Requester (signingService-FoxSigningRequester)

This component is a client to the **Fox Signing Transport** and provides a parent component with the ability to make requests to onboard with the Signing Service and to renew certificates.

It is not possible to drop an instance of the **Fox Signing Requester** onto any arbitrary component as some code is required to support the interaction between the component and the requester.



Cert Requester	Fox Signing Requester
Status	{ok}
Fault Cause	
Enabled	true
Requester State	Not Onboarded
Advance Renewal Percent	8 percent [1 - 100]
Advance Renewal	+000d 00h 00m
Retry Period	001d 00h 00m [5 minutes - 5 days]
Last Attempt	null
Last Success	null
Last Failure	null
Next Renewal Attempt	null
Alarm On Failure To Onboard	false
Alarm On Failure To Renew	true
Alarm Source Info	Alarm Source Info
Signing Service Station	Select Station...
Requester Id	99527568-d59e-4b4f-b8fd-b0c982ee6d2e

Property	Value	Description
Status	read-only	Displays the status of the requester component. It indicates if in alarm state when onboard/renew has failed.
Fault Cause	read-only	Displays the most recent fault message.
Enabled	true or false (defaults to true)	If false, actions are disabled.
Requester State	read-only	Displays one of the following states: `Not Onboarded, Approval In Progress, CSR Submitted, Renewal CSR Submitted, Renewal Failed, Onboarded`
Advance Renewal Percentage	read-only	Displays the percent of the signed certificates validity period in which a renewal will be automatically attempted. With a 100–day certificate and a value of 10%, renewal will be first tried with 10 days remaining.
Advance Renewal	read-only	Displays the period prior to certificate expiry in which a renewal will be attempted.
Retry Period	read-only	Displays the delay between automatic attempts to renew.
Last Attempt	read-only	Reports the last time an onboard/renew was attempted.
Last Success	read-only	Reports the last time a certificate was successfully retrieved.
Last Failure	read-only	Reports the last time a request failed.
Next Renewal Attempt	read-only	Reports the last time at which a renewal

Property	Value	Description
		will next be attempted.
Alarm on Failure to Onboard	True or false (defaults to false)	If true, an alarm is generated when an onboard attempt fails.
Alarm on Failure to Renew	True or false (defaults to true)	If true, an alarm is generated when a renewal attempt fails.
Alarm Source Info	additional properties	Provides configuration for alarms.
Signing Service Station	drop-down	Selects from a list of Niagara network stations the one containing the Signing Service.
Requester Id	read-only	Displays the unique Id for this requester as registered with the Signing Service. It is automatically populated.

Actions

Onboard: Registers this component with the Signing Service. The component will automatically check for approval and perform the CSR submission and retrieval.

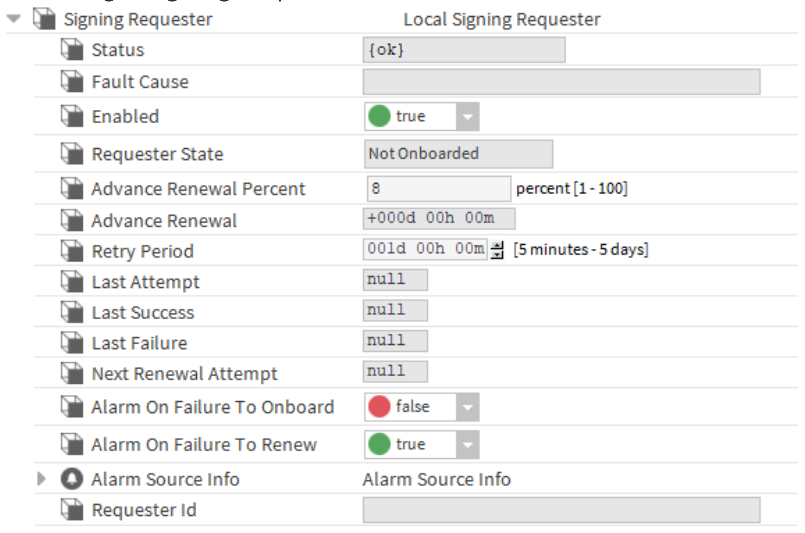
Renew: Forces a certificate renewal attempt.

Parent topic: Components

Local Signing Requester (signingService-LocalSigningRequester)

This component is a client to the Local Signing Transport and provides a parent component with the ability to make requests to onboard/renew with a Signing Service running in the same station.

It is not possible to drop an instance of the **Local Signing Requester** onto any arbitrary component as code is required to support the interaction between the component and the requester. For more information, see “*Installing a Signing Requester*”.



Signing Requester	Local Signing Requester
Status	{ok}
Fault Cause	
Enabled	● true
Requester State	Not Onboarded
Advance Renewal Percent	8 percent [1 - 100]
Advance Renewal	+000d 00h 00m
Retry Period	001d 00h 00m [5 minutes - 5 days]
Last Attempt	null
Last Success	null
Last Failure	null
Next Renewal Attempt	null
Alarm On Failure To Onboard	● false
Alarm On Failure To Renew	● true
Alarm Source Info	Alarm Source Info
Requester Id	

Property	Value	Description
Status	read-only	Displays the status of the requester component. It indicates if in alarm state when onboard/renew has failed.
Fault Cause	read-only	Displays the most recent fault message.
Enabled	true or false (defaults to true)	If false, actions are disabled.
Requester State	read-only	Displays one of the following states: `Not Onboarded, Approval In Progress, CSR Submitted, Renewal CSR Submitted, Renewal Failed, Onboarded`
Advance Renewal Percentage	read-only	Displays the percent of the signed certificates validity period in which a renewal will be automatically attempted. With a 100–day certificate and a value of 10%, renewal will be first tried with 10 days remaining.
Advance Renewal	read-only	Displays the period prior to certificate expiry in which a renewal will be attempted.
Retry Period	read-only	Displays the delay between automatic attempts to renew.
Last Attempt	read-only	Reports the last time an onboard/renew was attempted.
Last Success	read-only	Reports the last time a certificate was successfully retrieved.
Last Failure	read-only	Reports the last time a request failed.
Next Renewal Attempt	read-only	Reports the last time at which a renewal will next be attempted.
Alarm on Failure to Onboard	True or false (defaults to false)	If true, an alarm is generated when an onboard attempt fails.
Alarm on Failure to Renew	True or false (defaults to true)	If true, an alarm is generated when a renewal attempt fails.
Alarm Source Info	additional properties	Provides configuration for alarms.
Requester Id	read-only	Displays the unique Id for this requester as registered with the Signing Service. It is automatically populated.

Actions

Onboard: Registers this component with the Signing Service against the selected Profile. The component will automatically check for approval and perform the CSR submission and retrieval.

Renew: Forces a certificate renewal attempt.

Parent topic: Components

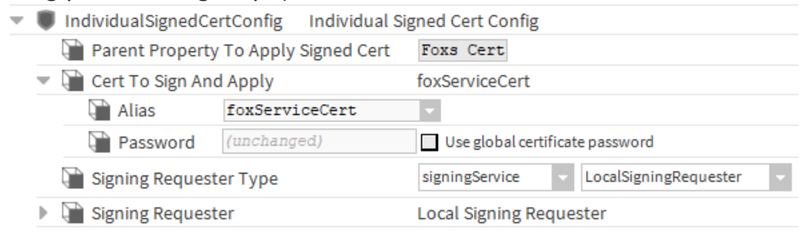
Individual Signed Cert Config (signingService-IndividualSignedCertConfig)

The **Individual Signed Cert Config** allows you to define the alias and password for the certificate that you use for the signing request.

The alias does not need to exist in the key store prior to onboarding. The component also offers the choice between using the **Fox Signing Requester** for when the **Signing Service** exists on a remote station, or the **Local Signing Requester** for when it exists in the same station. Optionally, you can add Certificate Parameters to this component to supply additional CSR parameters to those provided by the Signing Profile.

You can drop the **Individual Signed Cert Config** from the palette onto a specific component, such as the Fox Service, Web Service or Additional Https Cert. This will enable the parent component to make requests for an individual signed certificate from the Signing Service.

You can also install this component using a dedicated provisioning step (also see "*Installing Signing Requester using provisioning step*").



IndividualSignedCertConfig — Individual Signed Cert Config
Parent Property To Apply Signed Cert: Foxs Cert
Cert To Sign And Apply: foxServiceCert
Alias: foxServiceCert
Password: (unchanged) — Use global certificate password
Signing Requester Type: signingService — LocalSigningRequester
Signing Requester: Local Signing Requester

Property	Value	Description
Parent Property To Apply Signed Cert	read-only	Displays the name of the property on the parent (FoxService, WebService, and so on) to which the certificate alias and password will be written when the CSR has been signed.
Cert To Sign And Apply	additional properties	Specifies the alias and password of the certificate to use for storing the signed certificate. The alias does not need to exist in the key store prior to onboarding. A recommended alias will be populated by default.
Signing Requester Type	drop-down menu	Lists possible requester types.
Signing Requester	additional properties	Constitutes the requester component that will generate and submit the CSR.

Actions

Onboard: Registers this component with the Signing Service . The component will automatically check for approval and perform the CSR submission and retrieval.

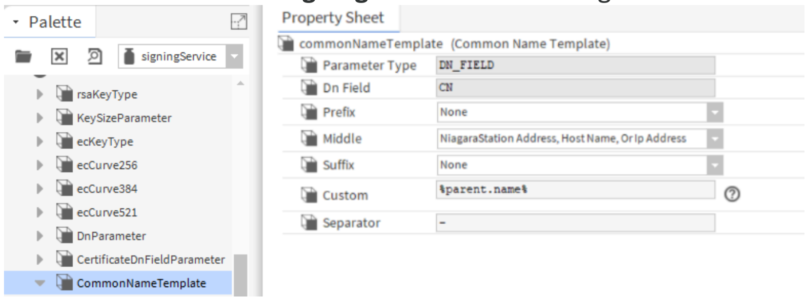
Renew: Forces a certificate renewal attempt.

Parent topic: Components

Common Name Template (platCrypto-CommonNameTemplate)

The **Common Name Template** component is a type of Certificate Parameter that will generate a dynamic common name for the signed certificate based upon configurable settings. The main benefit of this is the ability to generate a potentially unique common name for each component.

To use the **Common Name Template** component, drag it from the signingService palette beneath a Signing Profile underneath the **Signing Service** in a running station.



commonNameTemplate (Common Name Template)

Parameter Type	DN_FIELD
Dn Field	CN
Prefix	None
Middle	NiagaraStation Address, Host Name, Or Ip Address
Suffix	None
Custom	%parent.name%
Separator	-

The generated common name will by default resolve the Custom BFormat against the requesting component resulting in a common name matching the component's parent name.

This can be customized by modifying the Custom value, and by selecting values to prepend or append, or both in the respective Prefix and Suffix drop-downs. The Separator value is used to join the different parts of the common name.

Note: The NiagaraStation Address options for the Prefix, Middle, and Suffix properties (as of Niagara 4.14) will only resolve to the NiagaraStation Address when specified on a profile of the Signing Service, not when added to the remote signing requester. They also will not resolve if used with a local signing requester. In those cases, they will revert to the next fallback option. The reason is that a remote FOX connection is needed through which the Signing Service station that receives the request can locate the corresponding NiagaraStation in its NiagaraNetwork. This is not possible for local signing requesters or when the component is added to a remote station instead of the profile in the Signing Service station.

Available values for the three different parts of the name are:

- None — no value and no separator used
- Value of Format Slot — the value of the BFormat resolved against the requesting component
- Station Name
- Device Name — or an empty string if the component is not beneath a device
- Network Name — or an empty string if the component is not beneath a network
- Host Name
- Uid — a 16–character unique string
- Ip Address
- Host Name Or Ip Address
- NiagaraStation Address Or Host Name — This option will first attempt to locate a corresponding NiagaraStation in the NiagaraNetwork of the local station, which is the SigningService station, to find a match for the requester. If located, the Address property value of that NiagaraStation will be used. If not

found, it will use the Host Name of the requester.

- NiagaraStation Address Or Ip Address — This option will first attempt to locate a corresponding NiagaraStation in the NiagaraNetwork of the local station, which is the SigningService station, to find a match for the requester. If located, the Address property value of that NiagaraStation will be used. If not found, it will use the IP Address of the requester.

- NiagaraStation Address, Host Name, Or Ip Address — This option first will attempt to locate a corresponding NiagaraStation in the NiagaraNetwork of the local station, which is the SigningService station, to find a match for the requester. If located, the Address property value of that NiagaraStation will be used. If not located, it will next attempt to use the Host Name of the requester. If the Host Name is not available, it will use the IP Address of the requester.

Property	Value	Description
Parameter Type	read-only	Displays the name of the certificate parameter.
Dn Field	read-only	`CN`
Prefix	drop-down (defaults to None)	Selects the type of value to prepend to the common name. As of Niagara 4.14, the following additional options are available: • NiagaraStation Address Or Host Name • NiagaraStation Address Or Ip Address • NiagaraStation Address, Host Name, Or Ip Address
Middle	drop-down (defaults to NiagaraStation Address, Host Name, Or Ip Address)	Selects the type of value forming the middle of the common name. As of Niagara 4.14, the following additional options are available: • NiagaraStation Address Or Host Name • NiagaraStation Address Or Ip Address • NiagaraStation Address, Host Name, Or Ip Address
Suffix	drop-down (defaults to None)	Selects the type of value to append to the common name. As of Niagara 4.14, the following additional options are available: • NiagaraStation Address Or Host Name • NiagaraStation Address Or Ip Address • NiagaraStation Address, Host Name, Or Ip Address
Custom	BFormat string	Format used to resolve against the

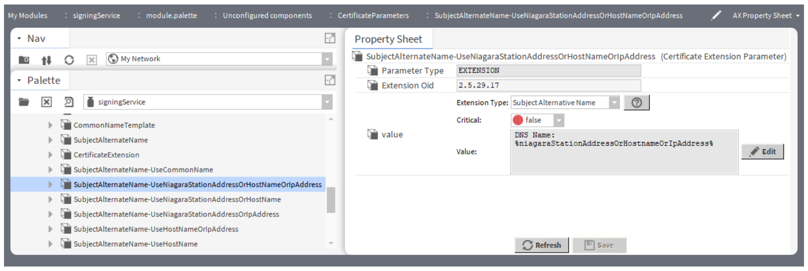
Property	Value	Description
		requesting component.
Separator	string	Specifies the value to join the different parts of the common name string.

Parent topic: Components

Certificate Extension Parameter (platCrypto-CertificateExtensionParameter)

The **Certificate Extension Parameter** component allows you to configure a Certificate Extension that will be applied to the CSR to be submitted for signing.

You can access this component from the signingService palette by expanding **Unconfigured components > CertificateParameters**, and add the desired Certificate Parameter component to a Signing Profile or a **Signed Cert Config** component. As of Niagara 4.14, the extension parameter “Subject Alternative Name” is added by default to the **serverProfile**.


My Modules
signingService
module.palette
Unconfigured components
CertificateParameters
SubjectAlternateName-UseNiagaraStationAddressOrHostNameOrIpAddress
AX Property Sheet
Nav
My Network
Palette
signingService
CommonNameTemplate
SubjectAlternateName
CertificateExtension
SubjectAlternateName-UseCommonName
SubjectAlternateName-UseNiagaraStationAddressOrHostNameOrIpAddress
SubjectAlternateName-UseNiagaraStationAddressOrHostName
SubjectAlternateName-UseNiagaraStationAddressOrIpAddress
SubjectAlternateName-UseHostNameOrIpAddress
SubjectAlternateName-UseHostName
Property Sheet
Edit
Refresh
Save

SubjectAlternateName-UseNiagaraStationAddressOrHostNameOrIpAddress (Certificate Extension Parameter)

Parameter Type EXTENSION
Extension Oid 2.5.29.17
value
Extension Type: Subject Alternative Name
Critical: false
Value:
```
DNS Name:
%niagaraStationAddressOrHostnameOrIpAddress%
```

Property	Value	Description
Parameter Type	read-only	Specifies the type of certificate parameter.
Extension Oid	string	Specifies the object Id of the certificate extension.
value	additional properties	Lists information that is configuration-specific to the selected type of extension. As of Niagara 4.14, the following added special format options are available: • %niagaraStationAddressOrHostnameOrIpAddress%: This option first will attempt to locate a corresponding NiagaraStation in the NiagaraNetwork of the local station, which is the SigningService station, to find a match for the requester. If located, the Address property value of that NiagaraStation will be used. If not located, it will next attempt to use the Host Name of the requester. If the Host Name is not available, it will use the IP Address of the requester. • %niagaraStationAddressOrHostname%: This option will first attempt to locate a corresponding NiagaraStation in the NiagaraNetwork of the local station, which is the SigningService station, to find a match for the requester. If located, the Address property value of that NiagaraStation will be used. If not found, it will use the Host Name of the requester. • %niagaraStationAddressOrIpAddress%: This option will first attempt to locate a corresponding

Property	Value	Description
		NiagaraStation in the NiagaraNetwork of the local station, which is the SigningService station, to find a match for the requester. If located, the Address property value of that NiagaraStation will be used. If not found, it will use the IP Address of the requester.

Parent topic: Components

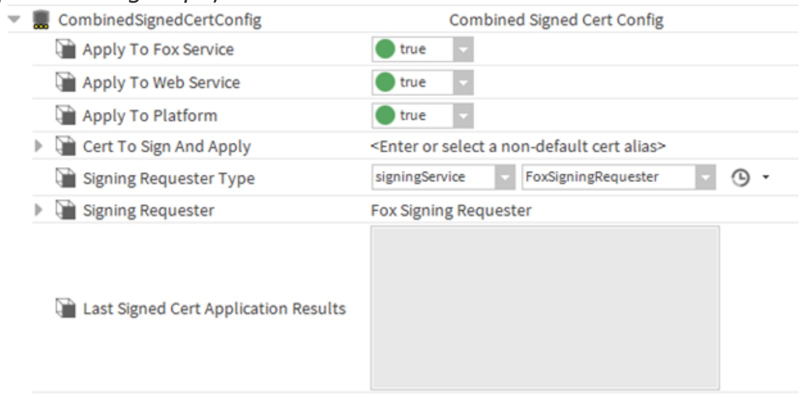
Combined Signed Cert Config (signingService-CombinedSignedCertConfig)

The **Combined Signed Cert Config** allows you to define the alias and password for the certificate that you use for the signing request.

The alias does not need to exist in the key store prior to onboarding. The component also offers the choice between using the Fox Signing Requester for when the Signing Service exists on a remote station or the Local Signing Requester for when it exists in the same station.

You can drop the **Combined Signed Cert Config** from the palette onto the station's Security Service allowing a single-signed certificate to be shared between any combination of the Fox Service/Web Service/Platform or all three.

You can install this component using a dedicated provisioning step (also see "*Installing Signing Requester using provisioning step*").



```
CombinedSignedCertConfig            Combined Signed Cert Config
  Apply To Fox Service              true
  Apply To Web Service              true
  Apply To Platform                 true
  Cert To Sign And Apply            <Enter or select a non-default cert alias>
  Signing Requester Type            signingService   FoxSigningRequester
  Signing Requester                 Fox Signing Requester
  Last Signed Cert Application Results
```

Property	Value	Description
Apply To Fox Service	true or false (defaults to true)	If true, the shared certificate will be used by the station's Fox Service.
Apply To Web Service	true or false (defaults to true)	If true, the shared certificate will be used by the station's Web Service.
Apply To Platform	true or false (defaults to true)	If true, the shared certificate will be used by the Platform's Web Service.
Cert To Sign And Apply	additional properties	Spefifies the alias and password of the certificate to use for storing the signed certificate. The alias does not need to exist in the key store prior to onboarding.

Property	Value	Description
Signing Requester Type	drop-down menu	Lists possible requester types.
Signing Requester	additional properties	Constitutes the requester component that will generate and submit the CSR.
Last Signed Cert Application Results	read-only	Displays a result summary of the last certificate signing process and to which services the cert was applied.

Actions

Onboard: Registers this component with the Signing Service . The component will automatically check for approval and perform the CSR submission and retrieval.

Renew: Forces a certificate renewal attempt.

Parent topic: Components

Plugins (views)

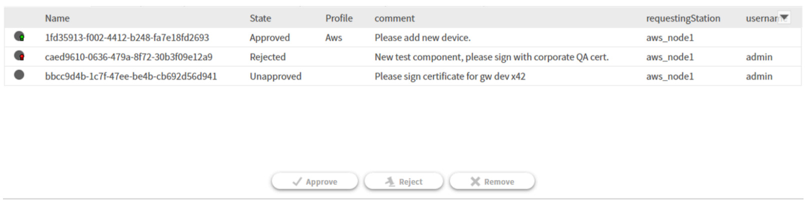
Plugins provide views of components and can be accessed in many ways. For example, double-click a component in the Nav tree to see its default view. In addition, you can right-click on a component and select from its **Views** menu.

- **Session Token Manager View (signingService-SessionTokenUxManager)**
  You can approve or reject tokens using actions on individual session tokens. Alternatively, you can use the **Session Token Manager** view, which is the default view for the store.

Parent topic: Components, views and windows

Session Token Manager View (signingService-SessionTokenUxManager)

You can approve or reject tokens using actions on individual session tokens. Alternatively, you can use the **Session Token Manager** view, which is the default view for the store.

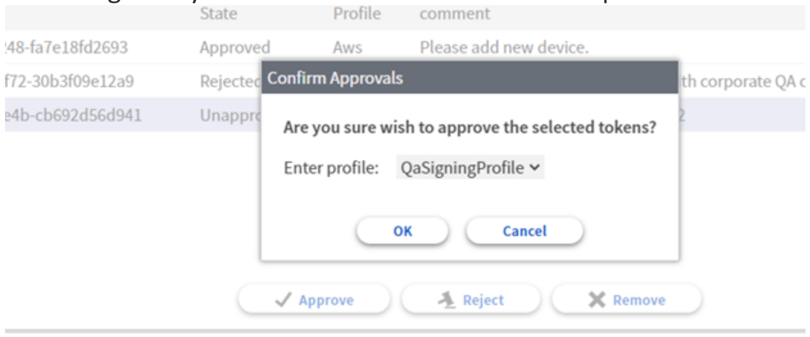


	Name	State	Profile	comment	requestingStation	usernar
●	1fd35913-f002-4412-b248-fa7e18fd2693	Approved	Aws	Please add new device.	aws_node1	
●	caed9610-0636-479a-8f72-30b3f09e12a9	Rejected		New test component, please sign with corporate QA cert.	aws_node1	admin
●	bbcc9d4b-1c7f-47ee-be4b-cb692d56d941	Unapproved		Please sign certificate for gw dev x42	aws_node1	admin

To access it, expand **Config > Services > SigningService > Transports > foxTransport** and double-click **Session Token Store**. The view displays each session token as a row that includes the status of the token plus all the metadata values submitted by the requesting component, which should help you on your decision to approve or reject the token.

Note: Only admin users are permitted to approve or reject tokens. They may batch approve, reject or delete session tokens by selecting multiple rows.

When approving a token, you must select the **Signing Profile** to which this request should be associated. The CSR will be signed by the CA certificate defined in that profile.



	State	Profile	comment
48-fa7e18fd2693	Approved	Aws	Please add new device.
f72-30b3f09e12a9	Rejected		th corporate QA c
e4b-cb692d56d941	Unappro		2

Confirm Approvals

Are you sure wish to approve the selected tokens?

Enter profile: QaSigningProfile

You can approve a rejected token at a later date provided the token has not automatically been removed before.

Parent topic: Plugins (views)

Certificate

A PKI (Public Key Certificate) or digital certificate is an electronic document used to prove ownership of a public key. The certificate includes information about the key, the identity of its owner, and the digital signature of an entity that verified the validity of the certificate’s contents. If the signature is valid, and the client can trust the signer, the client can be confident that it can use the public key contained in the certificate to communicate with the server.

Certificate Authority (CA)

An entity that issues the digital certificates used to certify the ownership of a public key by the named subject of the certificate. This allows system users to rely upon signatures or assertions made by the private key that corresponds to the certified public key. In this relationship model, the party that relies on the certificate trusts that the subject (owner) of the certificate is authentic because of the relationship of both parties to the CA.

Certificate Signing Request (CSR)

A block of encoded text that is given to a Certificate Authority (CA) when applying for an SSL certificate.

Distinguished Name (DN)

A string that uniquely identifies a certificate.

MQ Telemetry Transport (MQTT)

A protocol for exchanging queued messages between disparate systems.

Onboarding

Process of registering a requester with a service.

Requester/Requesting Component

A component that submits the Certificate Signing Request (CSR) to the service.

X509

A standard format of public key certificates.