appviewx

Citrix FAS - Integration

Introduction & Overview **3**

Need for Certificates and Certificate Types **4**

AppViewX CERT+ **4**

How AppViewX CERT+ Works 5

Citrix FAS integration with AppViewX **5**

Citrix FAS Installation and Configuration **7**

Prerequisites 7

Install FAS 7

FAS admin console and powershell 8

FAS admin console 8

FAS powershell cmdlets 9

Certificate templates 10

Citrix_RegisrationAuthority_ManualAuthorization 10

Citrix_RegisrationAuthority 11

Citrix_SmartcardLogon 12

Customizations 13

Turning off AD integration 13

FAS does not read the templates 15

Deploy templates to AD 15

Publish AD templates 16

Authorize FAS **16**

Online RA certificate request 16

Offline RA certificate request 17

Check the RA certificate 18

Configure FAS rule 20

Make a request for a user certificate 21

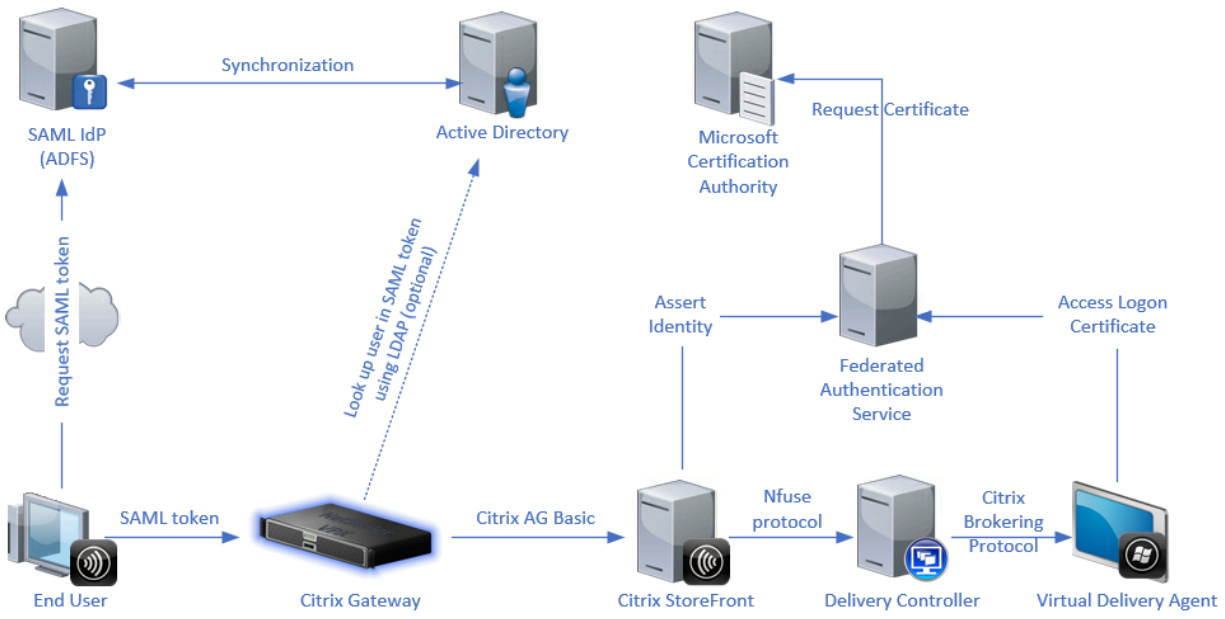
Check the user certificate.....27

Citrix FAS

Introduction & Overview

The Citrix Federated Authentication Service(FAS) is a privileged component designed to integrate with Active Directory Certificate Services. It dynamically issues certificates for users, allowing them to log on to an Active Directory environment as if they had a smart card. FAS allows StoreFront to use a broader range of authentication options, such as SAML (Security Assertion Markup Language) assertions. SAML is commonly used as an alternative to traditional Windows user accounts on the Internet.

The following diagram shows the integration of FAS with a certificate authority to provide services to StoreFront and XenApp and XenDesktop Virtual Delivery Agents (VDAs).



Trusted StoreFront servers contact the Federated Authentication Service (FAS) as users request access to the Citrix environment. The FAS grants a ticket that allows a single XenApp or XenDesktop session to authenticate with a certificate for that session. When a VDA needs to authenticate a user, it connects to FAS and redeems the ticket. Only

FAS has access to the user certificate's private key. The VDA sends each signing and decryption operation that it requires with the certificate to FAS.

Need for Certificates and Certificate Types

The FAS is authorized to issue smart card class certificates automatically on behalf of Active Directory users authenticated by StoreFront. This uses similar APIs to tools that allow administrators to provision physical smart cards.

When a user is brokered to a Citrix XenApp or XenDesktop Virtual Delivery Agent (VDA), the certificate is attached to the machine, and the Windows domain sees the logon as a standard smart card authentication.

Below are the three default certificate templates which the FAS publishes on AD:

- Smart Card Logon- (Requested by user every time he logs on to a VDA)
- Citrix_RegistrationAuthority
- Citrix_RegistrationAuthority_ManualAuthorization – These two certs Will be required only once during configuration, these Certificate Templates are required to authorize FAS as a certificate registration authority and will be renewed once in a year.

AppViewX CERT+

CERT+ is a ready-to-consume, scalable, and efficient certificate lifecycle management (CLM) solution to effectively automate and manage machine and application identities as an integral part of your cybersecurity strategy

By providing a unified process that weaves together different CAs, legacy on-prem and cloud services, next-gen technologies such as containers, IoT, and DevOps, AppViewX CERT+ effectively creates a CryptoMesh, providing a centralized control plane to automate enterprise-wide certificate lifecycle management and achieve crypto-agility.

How AppViewX CERT+ Works

- Certificate Discovery
- Holistic Certificate Visibility
- Prevent Certificate Related Outages
- Self-Service Certificate Automation
- Cloud Certificate Management
- Role-Based Access Control
- Reporting and Auditing
- End-to-End CLM Automation

As part of the product configuration, the following steps to be followed:

- Configure platform with required system details
    - Cloud Connector
    - SMTP Configuration
    - User and Certificate Groups
    - Users
- Configure Hydrant ID CA

Citrix FAS integration with AppViewX

When a user tries to logon to a VDA, a logon request will be sent to the FAS, after authenticating the user in AD, FAS will then connect with the AppviewX platform through a cloud connector which will also be attached to the same domain as of FAS, then the certificate issued is attached to the machine, and the Windows domain sees that logon as a standard smart card authentication.

Whenever Citrix FAS makes a CA Certificate or user certificate request to AppViewX cloud connector, the application receives the request in DCOM/DCERPC format, parses and analyze the request and pass it onto the application through regular REST API models for procuring certificate from the respective CA. The required certificate template or profile to be made available in the context of the CA for the AppViewX CERT+ to request the certificate in required format with all the EKU fields and KU fields. All the certificates issued are logged into the application and the

required expiry alert notification events can be configured. The devices will auto renew the certificate based on the threshold of the certificate.

Appendix

Citrix FAS Installation and Configuration

Prerequisites

- Windows domain with AD functional level >= Windows Server 2012 R2
- Windows server for installing FAS on (can be the domain controller) - Windows 2012 R2 and later
- The CItrix FAS machine and the CA (cloud connector) needs to be attached to the same domain.

Install FAS

Download FAS from https://www.citrix.com/downloads/federated-authentication-service

Copy the MSI to windows server, and double click to install it

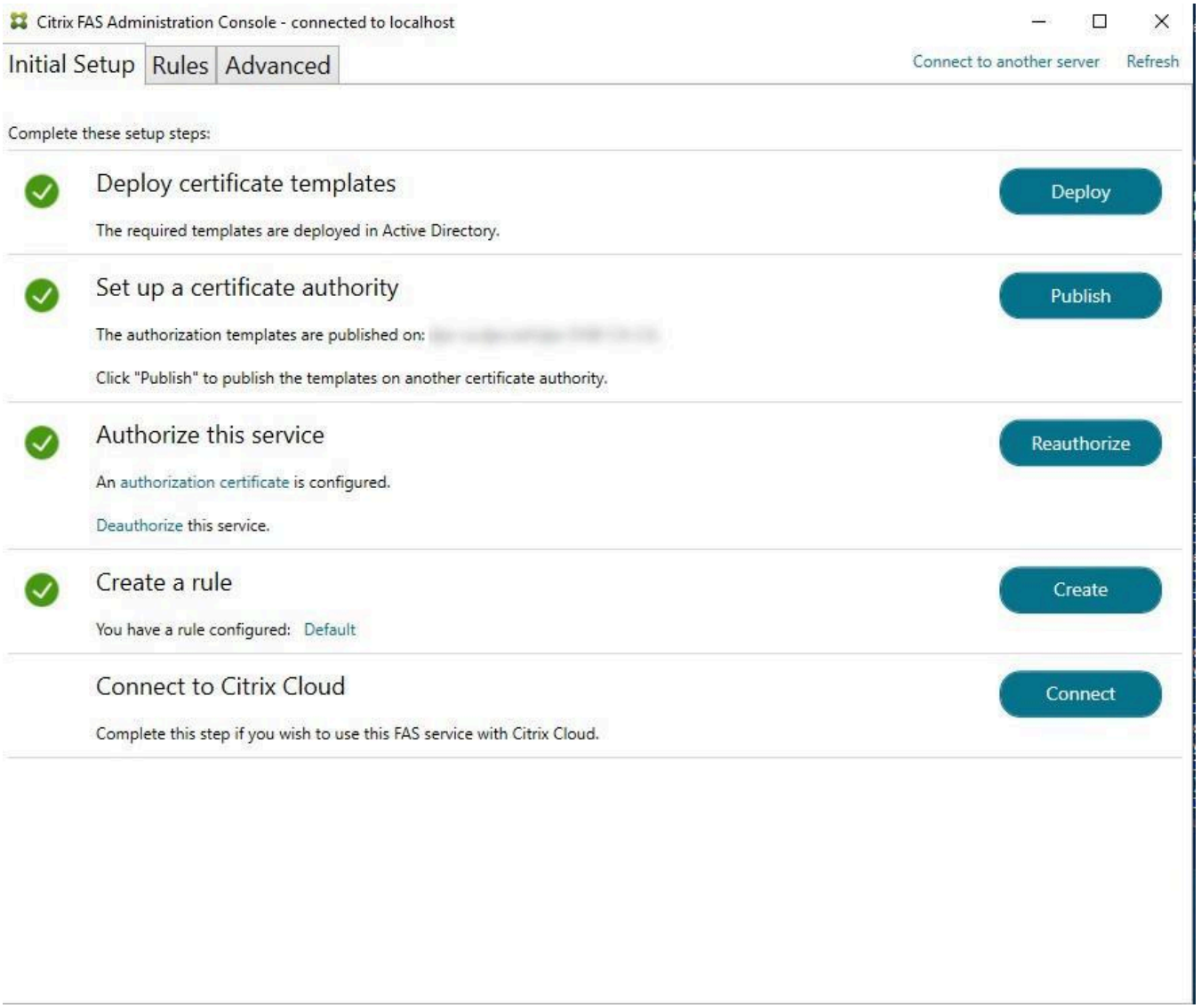
FAS admin console and powershell

You need to be running as a domain user who is a local admin (depending on your Windows settings, you may have to do "run as administrator...").

FAS admin console

Many of the steps here can be performed from the FAS admin console. This is a simple GUI that is sufficient for most customers' use cases. It is typically installed at:

```
C:\Program Files\Citrix\Federated Authentication Service\FasAdminConsole.exe
```



The GUI is "live" meaning that it polls FAS every 2 seconds to obtain its latest configuration - it can be helpful to leave the GUI open even when using powershell cmdlets.

Note however that the first two steps in the GUI (which involve communication with AD) are only updated if you click the refresh link at the top right of the GUI

FAS powershell cmdlets

All steps can be performed using powershell cmdlets supplied with FAS.

To use these, open a powershell window on the FAS server and type the following:

```
PS C:\> Add-PSSnapin ci*
PS C:\> Set-FasAdministrationPolicy -DefaultToLocalhost $true
```

The first command adds the cmdlets (you need to do this each time you open a new powershell window).

The second command sets a default target machine for commands, so that an address parameter does not have to be supplied in subsequent commands (you only need to do this once).

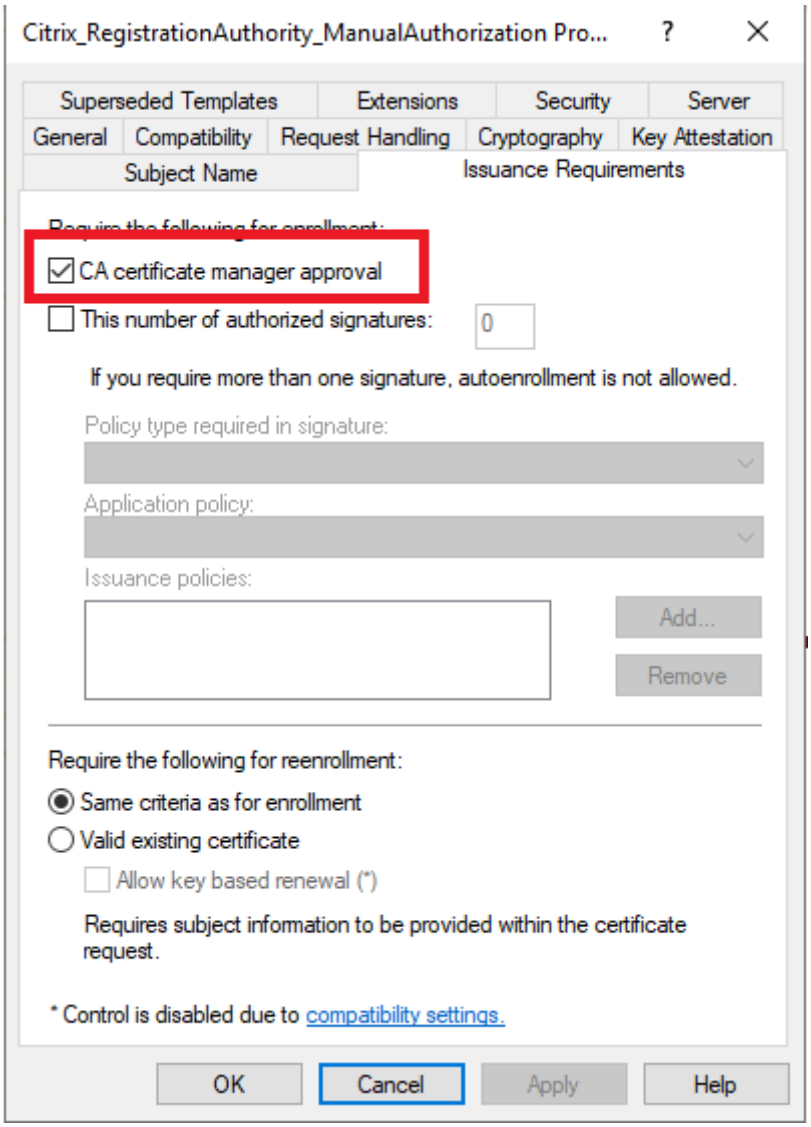
Certificate templates

FAS uses 3 certificate templates:

1. Citrix_RegistrationAuthority_ManualAuthorization
2. Citrix_RegistrationAuthority
3. Citrix_SmartcardLogon

Citrix_RegisrationAuthority_ManualAuthorization

This template requires the CA administrator's approval (this is the security step).

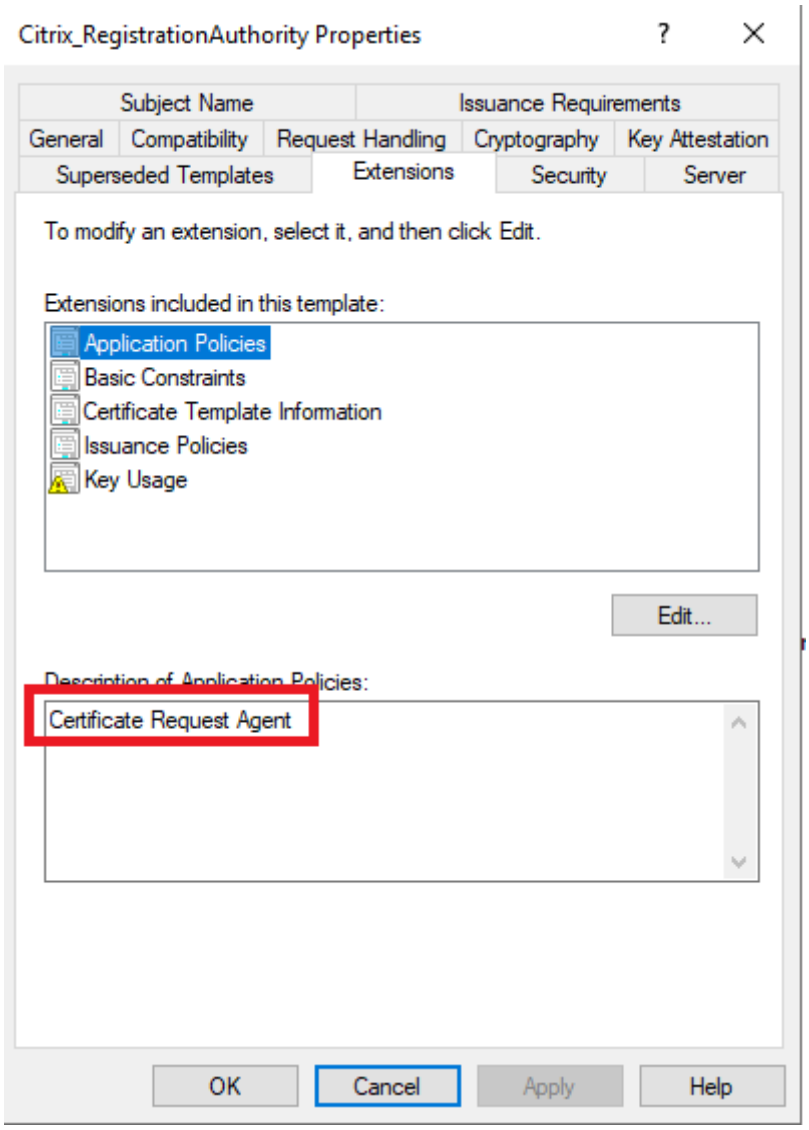


Once FAS has obtained a certificate with this template, it immediately uses it as authorisation to request a certificate with template Citrix_RegistrationAuthority.

The Citrix_RegisrationAuthority_ManualAuthorization certificate is then deleted.

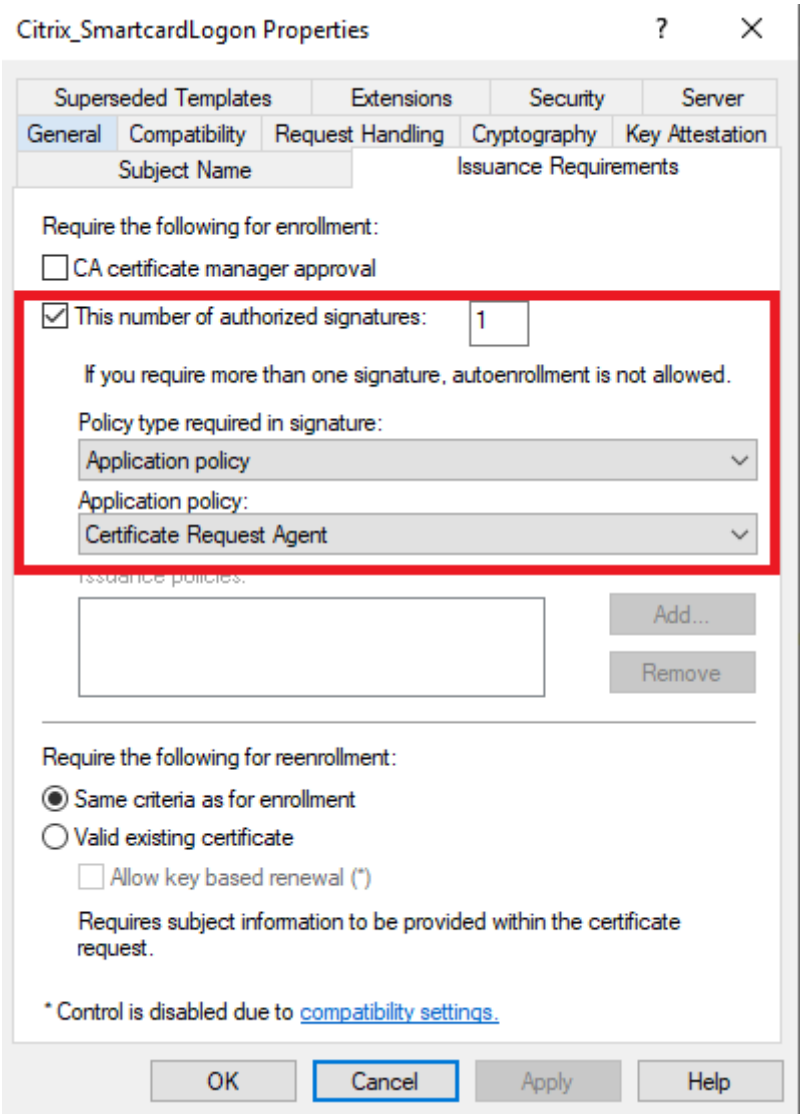
Citrix_RegisrationAuthority

This is the RA certificate template which authorizes FAS to act as an RA. It has the following EKU:



Citrix_SmartcardLogon

FAS uses this template to generate user certificates "on-the-fly" (so that FAS can perform single sign-on for the user). Its issuance requirements specify an RA certificate as authorisation:

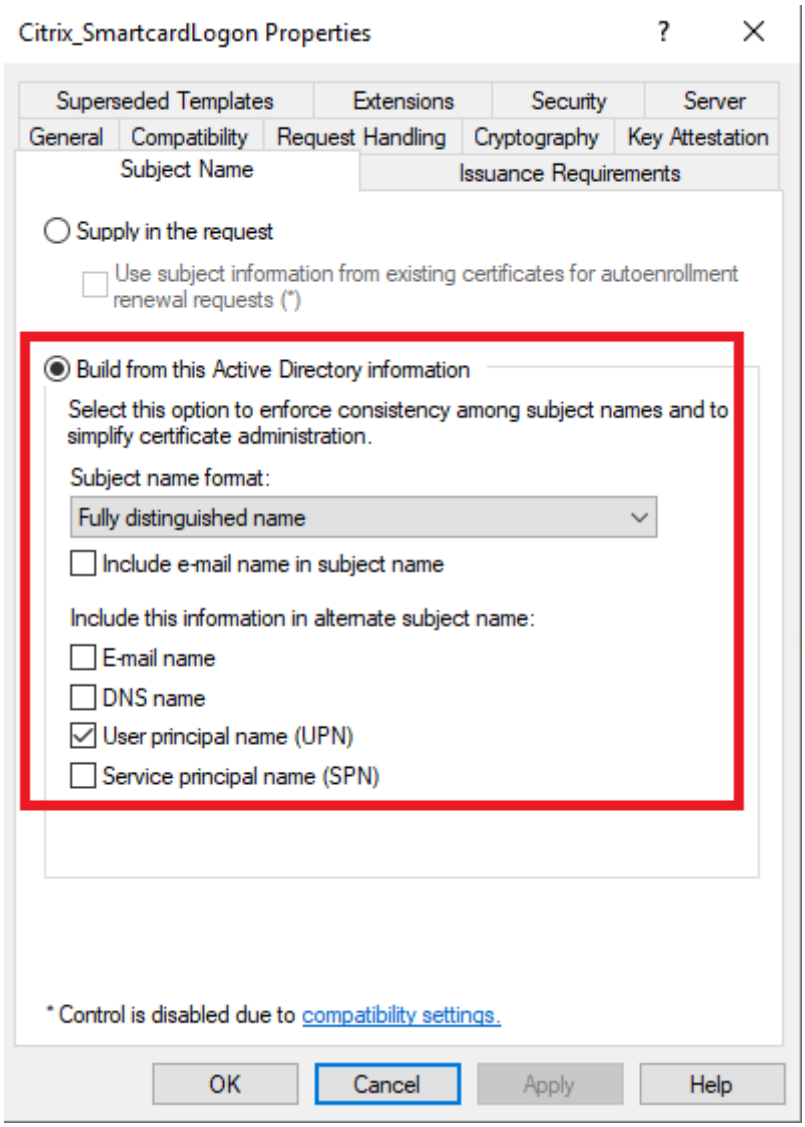


Customizations

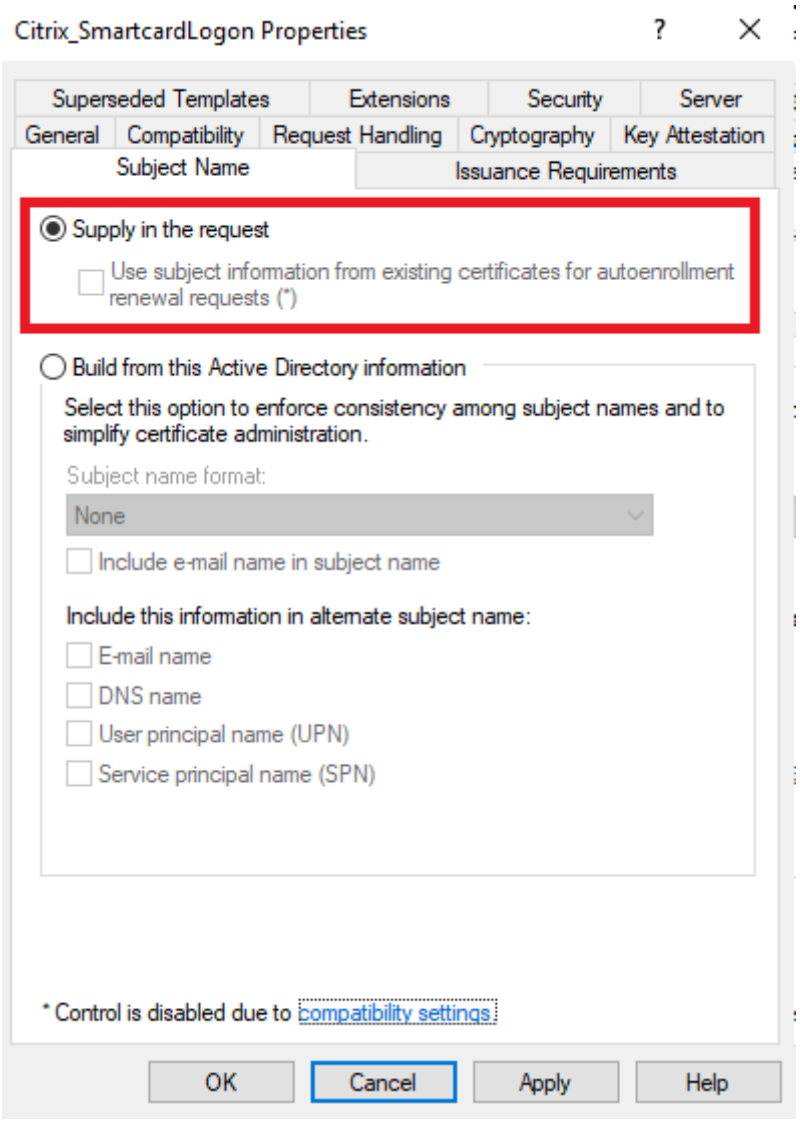
The templates can be customized, and it's also possible to configure FAS with an RA certificate without using these templates (see below).

Turning off AD integration

By default, the Citrix_SmartcardLogon template instructs the CA to query AD to populate fields in the certificate:



However, FAS supplies enough information in the certificate request that it's possible to change this setting to "Supply in the request". In this way, the CA would not need to query AD.



FAS does not read the templates

When creating a certificate request, the name of the template to use is part of the request.

Note however that FAS does **not** read the templates; for example, the key length to use is part of FAS's configuration; FAS does not read the minimum key length from the template.

Deploy templates to AD

In order to render the 3 templates usable from any CA, they must be deployed to AD.

Use the "Deploy" button in the FAS admin console (recommended).

Or alternately use powershell:

New-FasMsTemplate - deploy a template to AD Remove-FasMsTemplate - remove a template from AD The template files are typically located at:

```
C:\Program Files\Citrix\Federated Authentication Service\CertificateTemplates
```

Publish AD templates

This step makes a template available for use from a Microsoft CA / PKIaaS / Hydrant Id CA, and therefore may require a different procedure for a 3rd party CA.

You can use the CA's GUI to publish the template, or **use the "Publish" button in the FAS admin console (recommended)**.

Or use powershell:

Publish-FasMsTemplate Unpublish-FasMsTemplate

Authorize FAS

The goal is for FAS to be configured with an RA certificate.

From the FAS admin console, you can use the "Authorize" button (and subsequently remove authorisation using the deauthorize link). The CA administrator has to approve the request. FAS polls the CA awaiting a response, and the FAS admin console shows a spinner while the request is still pending. Using powershell is more flexible (see below).

Online RA certificate request

The same functionality that the FAS admin console performs can be achieved with these powershell cmdlets:

New-FasAuthorizationCertificate

Remove-FasAuthorizationCertificate Get-FasAuthorizationCertificate e.g..:

```
PS C:\> $CA =
 (Get-FasMsCertificateAuthority)[0].Address PS C:\>
 $CA
ca.example.com\example-ca
PS C:\> New-FasAuthorizationCertificate -CertificateAuthority $CA -CertificateTemplate
 Citrix_RegistrationAuthority -AuthorizationTemplate
 Citrix_RegistrationAuthority_ManualAuthorization

Id :
 e1894aaf-a792-4793-873c-c4f64ecf4ff5
 Address : ca.example.com\example-ca
 TrustArea :
CertificateRequest :
Status : WaitingForApproval

PS C:\> Remove-FasAuthorizationCertificate -Id e1894aaf-a792-4793-873c-c4f64ecf4ff5
```

Both Citrix_RegistrationAuthority_ManualAuthorization and Citrix_RegistrationAuthority (or similar) templates are involved in the above. The CA administrator has to approve the request. FAS polls the CA awaiting a response. You can use the FAS admin console GUI or the Get-FasAuthorizationCertificate cmdlet to track the progress of the pending CSR.

Offline RA certificate request

You can use powershell to create a CSR for the RA certificate, and then import the CSR response.

It's up to the administrator to supply the generated CSR to the CA, get the response, and import it into FAS.

New-FasAuthorizationCertificateRequest - generate a PKCS#10 request

Import-FasAuthorizationCertificateResponse - import a PKCS#7 response Templates are not directly involved.

```
PS C:\> New-FasAuthorizationCertificateRequest

Id :
 86d516bf-79ee-4070-921c-6ae3ec44e79f
 Address : [Offline CSR]
TrustArea :
CertificateRequest : -----BEGIN CERTIFICATE-----
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
 AA
 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
 AAA
 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
 AAA
 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
 AAA
 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
 AAA
 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
 AAA
 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
 AAA
 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
 AAA
 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
 AAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=
-----END CERTIFICATE-----
Status : WaitingForApproval

PS C:\> Import-FasAuthorizationCertificateResponse -Id 86d516bf-79ee-4070-921c-6ae3ec44e79f
 - Pkcs7CertificateFile C:\temp\response.p7b

Id :
 86d516bf-79ee-4070-921c-6ae3ec44e79f
 Address : [Offline CSR]
TrustArea :
 3cf10c59-89ee-4e90-93a6-d81aa161449c
 CertificateRequest :
Status : Ok
```

Check the RA certificate

Once you have configured FAS with an RA certificate using one of the above methods, you can inspect it to ensure it has the correct attributes.

To see the certificate, **use the FAS admin console (recommended)** - in the "Authorize this service" section, click the "An authorization certificate is configured" link.

Alternatively, you can use powershell:

Get-FasAuthorizationCertificate

For example:

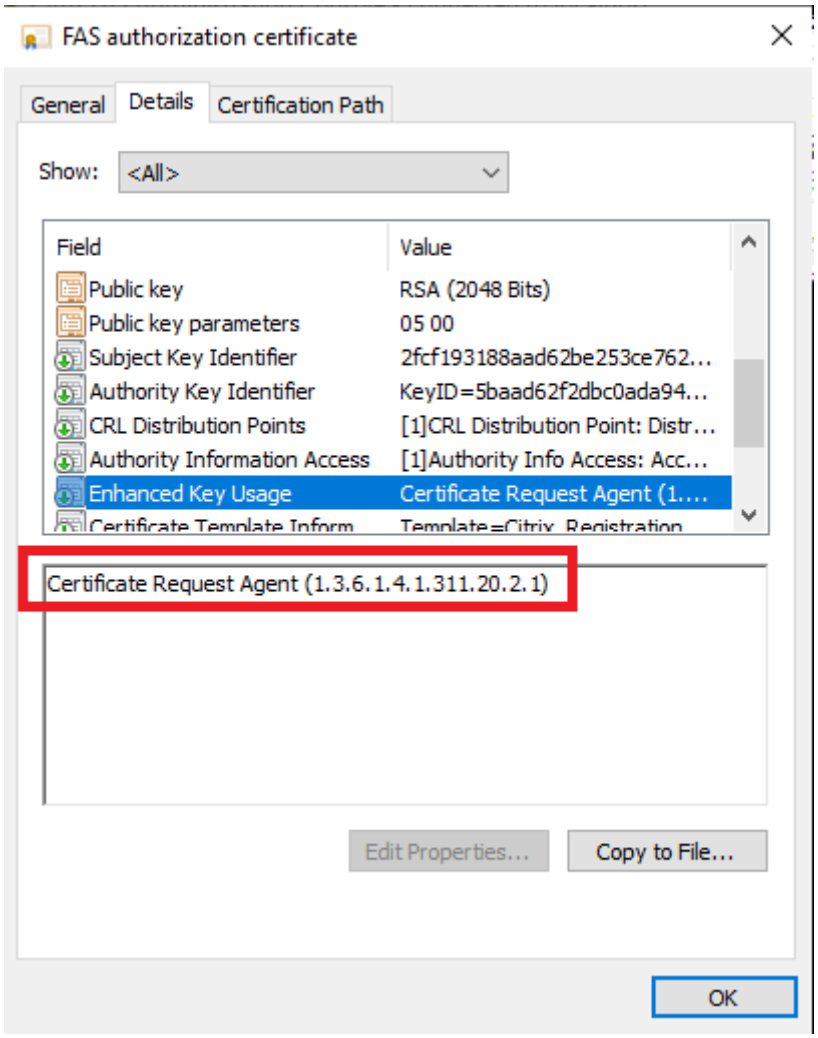
```
PS C:\> Get-FasAuthorizationCertificate -FullCertInfo

ThumbPrint    : 9676FE1C7CE816CE6F71822F8809EEF51CC1FFD7
ExpiryDate    :     01/07/2023
15:01:38 Certificate     :
-----BEGIN CERTIFICATE-----
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=
=
-----END CERTIFICATE-----


```
PrivateKeyIdentifier                   :
02733f3b-ca1e-4bdb-b646-1696166715d7
PrivateKeyProvider  :         Microsoft
Software     Key     Storage     Provider
PrivateKeyIsCng      : True
Id               : 49fa5bbd-71c2-4d7a-b27d-581e65617555
Address          : ca.example.com\djw-DJW-CA-CA
```

```
TrustArea      :
d5b119fc-f7ab-46ec-a743-212a0da979df
CertificateRequest  :
Status         : Ok
```

Check that the RA certificate has the required EKU:



Configure FAS rule

Once you have configured FAS with an RA certificate, this step adds the rest of the FAS configuration.

It's probably easiest to use the FASadmin console to do this: Click the "Create" button

A wizard appears; for most screens just click "Next" A couple of the screens pertain to CA-related stuff: the template that FAS will use to request user certificates the CA(s) FAS will contact to request a user certificate

Depending on CA, and how closely it integrates with AD, it may or may not appear in the CAs presented. If your CA does not appear, you may wish to use powershell instead.

Note that the CA you configure FAS to contact does not have to be the same CA that issued the RA certificate.

Powershell

The relevant commands are:

New-FasRule

Get-FasRule

Set-FasRule

Remove-FasRule

New-FasCertificateDefinition

Get-FasCertificateDefinition

Set-FasCertificateDefinition

Remove-FasCertificateDefinition

For example:

```
PS C:\> $RaCertInfo = (Get-FasAuthorizationCertificate)[0]
PS C:\> New-FasCertificateDefinition -Name Default_Definition -CertificateAuthorities "ca.example.com\example-
 ca" -MsTemplate "Citrix_SmartcardLogon" -AuthorizationCertificate $RaCertInfo.Id
PS C:\> New-FasRule -Name Default -CertificateDefinitions Default_Definition -StoreFrontAcl "O:BAG:DUD:P(D;OICI;
 CC;;;DC)"
```

Modify to specify your CA (i.e. the CA FAS will contact to make a certificate request) and the template that FAS will specify in the certificate request.

Make a request for a user certificate

The following powershell can be used to make FAS request a new user certificate, just as would happen

when FAS is deployed in a real environment. Crucially, **the user certificate must be issued by the CA automatically** e.g. without any manual approval from the CA administrator.

Use the CSR test cmdlet

Use powershell to instruct FAS to create a certificate request, send it to the CA, and get the response:

```
PS C:\> Test-FasCertificateSigningRequest -UserPrincipalName "user@example.com" -Rule
default

Description Succeeded FailureDetail TestInfo
----------- --------- ------------ --------
CSR         T                      {[UserPrincipalName, user@example.com], [SecurityContext, ],
default],   r                      [RuleName,
[Ce...      u
            e
```

(Use a UPN from your AD).

If the CSR fails, you can drill down into the FailureDetail:

```
PS C:\> $testResult = Test-FasCertificateSigningRequest -UserPrincipalName
"user@example.com" -Rule default PS C:\> $testResult
 ---------- -------- -----------
Description Succeeded FailureDetail

CSR        False An exception occurred: System.Exception: The CSR failed:
CCertRequest::Submit: The RPC server ...


PS C:\> $testResult.TestInfo

Key                     Value
---                     -----
UserPrincipalName       user@example.com SecurityContext
RuleName                default
CertificateDefinitionNameDefault_Definition CertificateAuthorityAddress
ca.example.com\example-ca SamAccountName EXAMPLE\user
CertificateTemplateName Citrix_SmartcardLogon PolicyOids
AuthorizationCertificateThumbprint
2C2DACEA44DCD88978BE2965DD86489FC1573613 AuthorizationCryptoProviderName
Microsoft Software Key Storage Provider AuthorizationPrivateKeyIdentifier
3406c405-22b2-421d-ab8c-4d1343934710 CryptoProviderName  Microsoft
Software Key Storage Provider
PrivateKeyIdentifier    80731419-5cd4-4916-a286-2f17e2e6383f


PS C:\> $testResult.FailureDetail
An exception occurred: System.Exception: The CSR failed:
CCertRequest::Submit: The RPC server is unavailable. 0x800706ba (WIN32:
1722 RPC_S_SERVER_UNAVAILABLE) --->
System.Runtime.InteropServices.COMException: CCertRequest::Submit: The RPC
server is unavailable. 0x800706ba (WIN32: 1722 RPC_S_SERVER_UNAVAILABLE)
  at CERTCLILib.ICertRequest3.Submit(Int32 Flags, String strRequest,
  String strAttributes, String strConfig) at
```

```
  Citrix.Authentication.UserCredentialServices.PkiCore.CertificateIssuance
  .MicrosoftCertificateAuthority.
SubmitCertificateRequest(String subjectSamAccount, Request request,
IList`1 optionalExtensions)
  at
Citrix.Authentication.UserCredentialServices.Service.Diagnostics.Diagnosti
cPkiProvider.PerformCsr(String upn, IPrivateKey userKey, String
certificateAuthorityAddress, ITrustArea caCredentials, String
templateName, String securityContext, String[] policyOids)
  at
Citrix.Authentication.UserCredentialServices.Server.FasDiagnostics.TestCer
tificateSigningRequestImpl (DiagnosticTestResult testResult, String upn,
String securityContext, String ruleName, Boolean reuseCachedTestKey,
String certificateDefinitionName, String certificateAuthorityAddress)
  --- End of inner exception stack trace ---
  at
Citrix.Authentication.UserCredentialServices.Server.FasDiagnostics.TestCer
tificateSigningRequestImpl (DiagnosticTestResult testResult, String upn,
String securityContext, String ruleName, Boolean reuseCachedTestKey,
String certificateDefinitionName, String certificateAuthorityAddress)
  at
  Citrix.Authentication.UserCredentialServices.Server.FasDiagnostics.<>c
  DisplayClass8_0.
<TestCertificateSigningRequest>b_0(DiagnosticTestResult testResult)
  at
Citrix.Authentication.UserCredentialServices.Service.Diagnostics.Diagnosti
cUtils.RunTest[T](Action`1 testAction)
```

Note that if successful, FAS does not retain the generated certificate.

Use the new user certificate cmdlet

These commands are less helpful in terms of error reporting, but the generated certificate is retained by FAS, so

you can inspect it. New-FasUserCertificate

Remove-FasUserCertificate Get-FasUserCertificate

Use Remove-FasUserCertificate first, to clear out any cached certificates, for example:

```
PS C:\> Remove-FasUserCertificate
PS C:\> New-FasUserCertificate -UserPrincipalName "user@example.com" -Rule
default -CertificateDefinition default_definition


ThumbPrint      : F2B69CE11E2B23DACD499E13EE74907E42263ADF
UserPrincipalName    : user@example.com Role     : default
CertificateDefinition : default_definition SecurityContext     :
ExpiryDate      : 08/07/2021 15:14:10
Certificate     : -----BEGIN CERTIFICATE-----
```

```
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

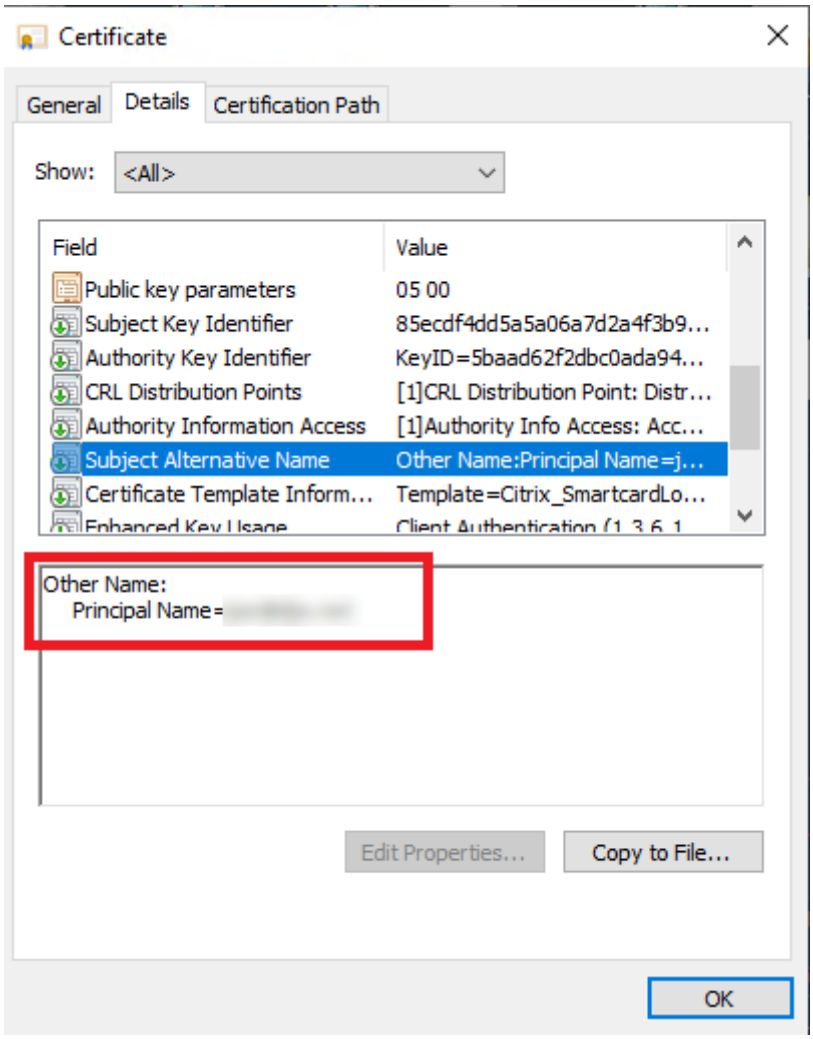
```
AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==
-----END CERTIFICATE-----
```

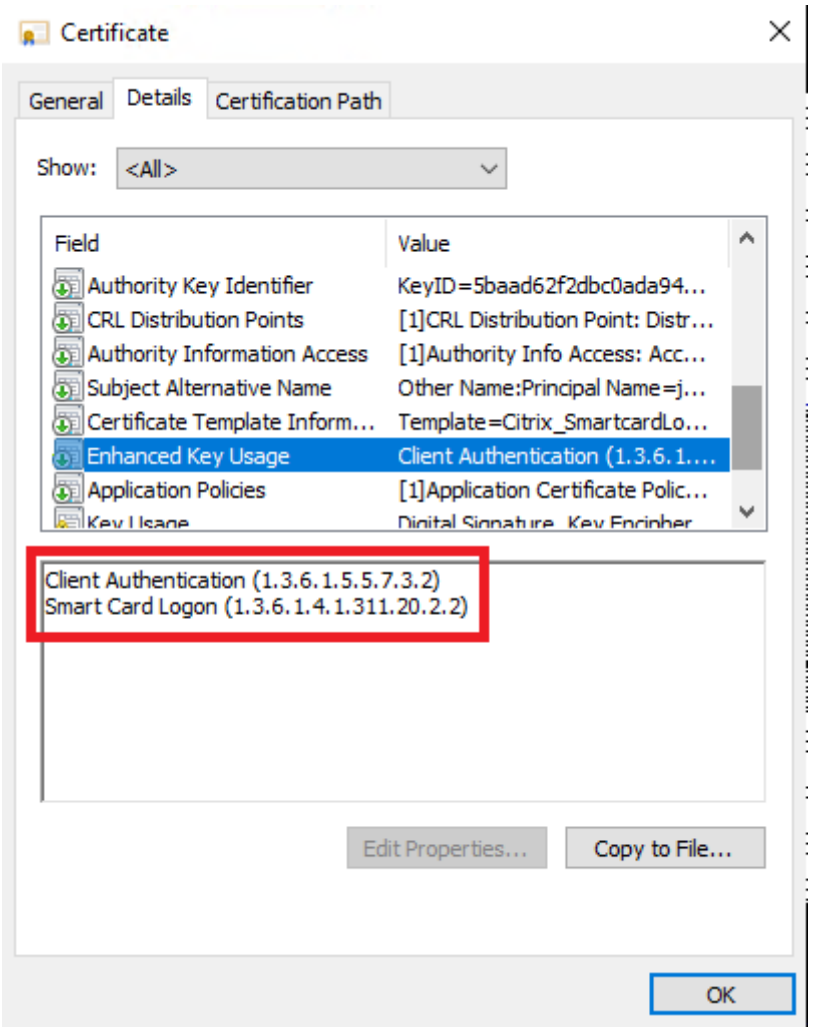

Note that New-FasUserCertificate has the following properties:

- if there is already a valid certificate cached for the given UPN, it simply returns that certificate, and doesn't generate a new one
- if there is a problem generating a certificate, the cmdlet does not return anything; there will however be entries in the windows application event log indicating the problem

Check the user certificate

Inspect the issued user certificate. It should have the user's UPN in the SAN:





Contact Information

AppViewX, Inc.

New York (HQ). City Hall,

222 Broadway New York, NY 10038

Tel: +1 (212) 400 7541

Tel: +1 (929) 955 0055

Email: info@appviewx.com

Web: http://www.appviewx.com