Active Directory Certificate Services

Certificate Request Workflow- Quick Start Manual

Version

1-Mar-24

Version 1.1

Prepared by

Bulgwei

Table of Contents

Table of Contents

Table of Contents **2**

Introduction **3**

Configuration 3

Installation **4**

Prerequisites 4

Computer hosting the Agents 4

Agent Account 4

Working Folder Structure 4

CA Permissions 5

How to configure the Automatic Request Processing Agents 6

Defining the Config.xml 6

Installing the Agents 8

Request Processing Workflow **9**

Incoming Requests 9

Submitted Requests 10

Archiving the Requests 11

Request Clean Up 11

The Control File 12

Process Status Information 13

Retrieving the Enrolled Certificates 15

Uninstalling the Agents 16

Updating the Agents Configuration 16

Pre-validating incoming requests 17

Appendix **18**

Arguments accepted by the installation script (Install-RequestAgnts.ps1) 18

Fixed event IDs of the agents 18

CSR structure for pre-validation 19

Introduction

Certificate Request Processing provides a secure way of submitting requests and and separately additional Subject Alternative Names (SANs) to the specified online CA. The workflow consists of two scripts that will run periodically as scheduled tasks, referred to as **Agents** in this document.

- Submit Agent (SubmitAgnt.ps1)

  The submit agent is looking into a filesystem (CIFs)-based inbox and submits the content including additional SANs to the CA. At the CA the requests stay pending and waiting for approval of a Certificate Manager. During the pending state, possible additional SANs will be securely added to the request (without having EDITF_ATTRIBUTESUBJECTALTNAME2 on the policy module enabled).

- Enroll Agent (EnrollAgnt.ps)

  The enroll agent verifies if a request has been issued/approved and exports the appropriate certificate onto a filesystem (CIFs)-based outbox.

The installation of the agents and their configuration is based on the XML-configuration file.

Configuration

Review the ***config.xml*** file to assign the necessary configuration that fits to the network and account requirements:

> **Note:** You should use an **XML editor with syntax highlighting** to edit the configuration file.

Installation

Prerequisites

- You will need Windows **PowerShell Version 4.0** or newer installed on your machine.
- You need to be **local administrator** to run the installation of both agents.

Computer hosting the Agents

Best practice is to install the agents directly on the issuing CAs.

Agent Account

You need to define the account in which context the agents will run and the names for the tasks. A **Group managed service account is recommended**. The account to be used must be defined in the config.xml, details will be provided in the “Defining the config.xml” section of this document.

Ensure that if using a group managed service account, the account have been assigned to the systems where you want to install the agents on. Additionally, ensure that the used account have appropriate logon permissions.

The account needs the following logon permissions:

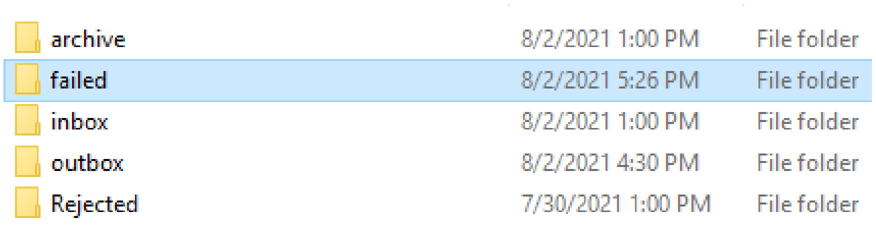
- Group managed service account – Log on as a service
- ”normal” user-based service account – Log on as a batch job

Working Folder Structure

To make the process working, a special working folder structure is necessary in the network. The base folder is defined in the **BaseDir** node in the config.xml:

```
<!--
    Description:
        The BaseDir Node specifies the base folder in the network where
        the script is looking for incoming requests
-->
<BaseDir>\\san.fabrikam.com\sanshare\AutoReqCntl\FabIssuingCa1</BaseDir>
```

The sub folder structure must be created manually.



```
archive        8/2/2021 1:00 PM     File folder
failed         8/2/2021 5:26 PM     File folder
inbox          8/2/2021 1:00 PM     File folder
outbox         8/2/2021 4:30 PM     File folder
Rejected       7/30/2021 1:00 PM    File folder
```

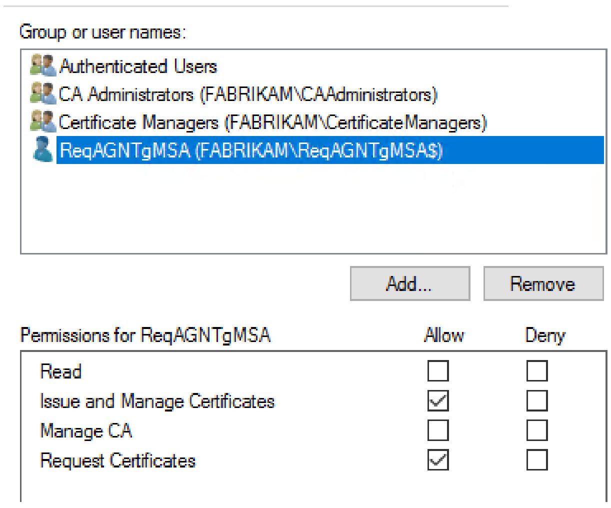
All folders should be owned by the account which runs the agents and only contributing accounts (sending request into the inbox) should have write permissions to the inbox folder. For the outbox folder, only read permission are acceptable for users. The folders “archive”, “failed” and “rejected” should be prevented from user access

.

CA Permissions

The agent account must have the following permissions on the submitting CA:

- Request Certificates
- Issue and Manage Certificates



Group or user names:

- Authenticated Users
- CA Administrators (FABRIKAM\CAAdministrators)
- Certificate Managers (FABRIKAM\CertificateManagers)
- ReqAGNTgMSA (FABRIKAM\ReqAGNTgMSA$)

Permissions for ReqAGNTgMSA	Allow	Deny
Read	☐	☐
Issue and Manage Certificates	✓	☐
Manage CA	☐	☐
Request Certificates	✓	☐

How to configure the Automatic Request Processing Agents

Defining the Config.xml

Agent Account

You need to define the **account in which's context the agents** will run and the names for the Tasks. The created tasks will run with a timely offset of 30 minutes against each other with a defined repetition interval.

```
<Install>
    <!-- define the account that should be used for events (gMSA has to end with a "$") -->
    <AgentAccountName>fabrikam\ReqAGNTgMSA$</AgentAccountName>

    <!-- define the task name for the "Submit Agent" -->
    <SubmitTaskName>SubmitAgent</SubmitTaskName>

    <!-- define the task name for the "Enroll Agent" -->
    <EnrollTaskName>EnrollAgent</EnrollTaskName>

    <!-- define the task repetition intervals in hours -->
    <TaskRepetitionInterval>1</TaskRepetitionInterval>
</Install>
```

Eventlog and Event Source

You must define the **eventlog and event source** that should be used for status messages. It will be created as well during the installation.

```
<Eventlog>
    <!-- define the event log that should be used for events
        MANDATORY! MUST BE DEFINED!
     -->
    <EventLog>ADCS-Scripts</EventLog>

    <!-- define the event source that should be used for events in the eventlog
        MANDATORY! MUST BE DEFINED!
    -->
    <EventSource>ADCS-Enroll-Agent</EventSource>
```

> **Note:** If you don't use a Group Managed Service Account, you might need to enter the account's password manually to the created tasks.

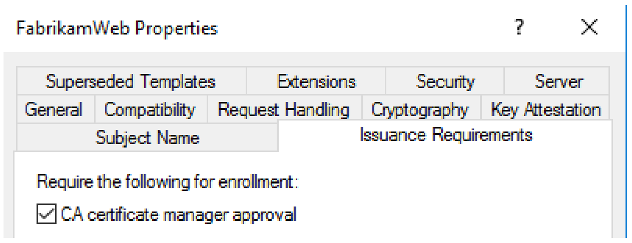
Certificate Templates

Each certificate template that can be used by the agents has to be added into the config XML file:

```
<!--
    Description:
        The Template Node specifies the allowed certificate templates for the request automation
-->
<Template>FabrikamWeb</Template>
<Template>FabrikamWeb2</Template>
<Template>FabrikamWeb3</Template>
<!--
    Description:
        The DefaultTemplate Node specifies the certificate template to be used if no template has been specified
-->
<DefaultTemplate>FabrikamWeb</DefaultTemplate>
```

Additionally, a default certificate template must be configured. That is used in case no explicit template has been provided by an incoming request.

All templates that should be used for this workflow must be configured to pend the request on the CA:

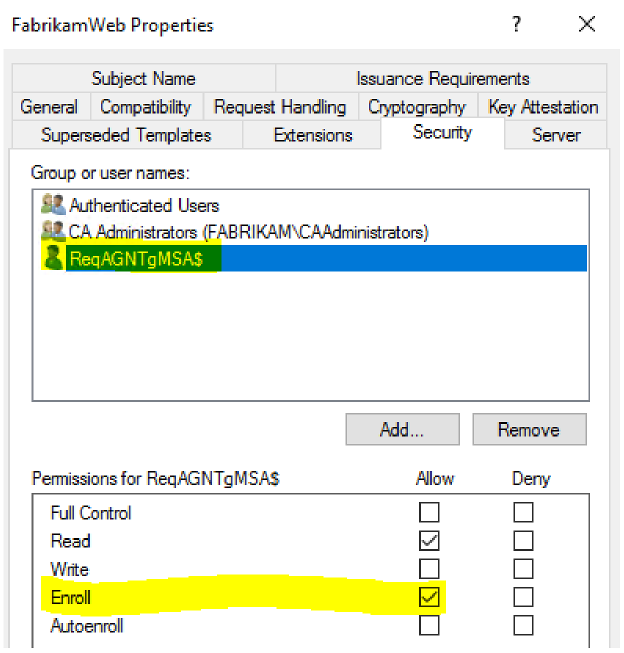

FabrikamWeb Properties

Superseded Templates | Extensions | Security | Server
General | Compatibility | Request Handling | Cryptography | Key Attestation
Subject Name | Issuance Requirements

Require the following for enrollment:

☑ CA certificate manager approval

The account who runs the agents, must have enroll permissions on the configured certificate templates:


FabrikamWeb Properties

Subject Name | Issuance Requirements
General | Compatibility | Request Handling | Cryptography | Key Attestation
Superseded Templates | Extensions | Security | Server

Group or user names:

- Authenticated Users
- CA Administrators (FABRIKAM\CAAdministrators)
- ReqAGNTgMSA$

Add... | Remove

Permissions for ReqAGNTgMSA$	Allow	Deny
Full Control	☐	☐
Read	☑	☐
Write	☐	☐
Enroll	☑	☐
Autoenroll	☐	☐

Certification Authority

The CaName Node specifies the target CA for the request automation. The name must be provided in format "fqdn\CA name".

Mis. Settings

Also reviewing the other configuration nodes is highly recommended!

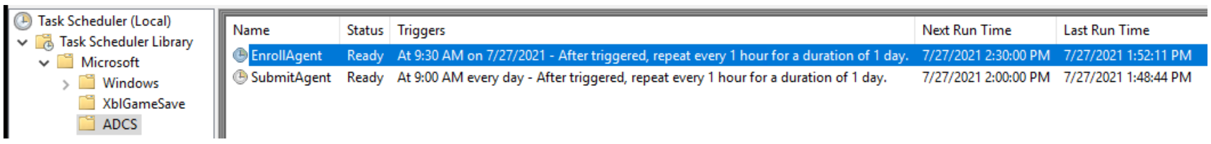
Installing the Agents

After the initial configuration has been defined, install the Agents by running:

```
Install-RequestAgnts.ps1 -InstallAgents
```

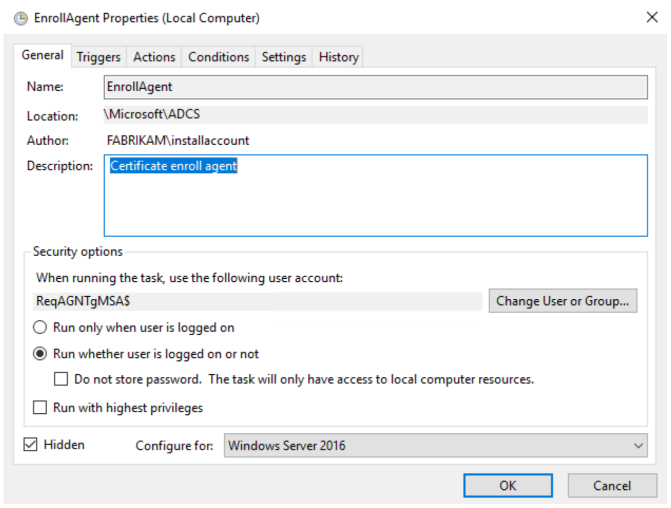
The installation routine will

- create and configure scheduled tasks for each agent.



Task Scheduler (Local)
- Task Scheduler Library
  - Microsoft
    - Windows
    - XblGameSave
    - ADCS

Name	Status	Triggers	Next Run Time	Last Run Time
EnrollAgent	Ready	At 9:30 AM on 7/27/2021 - After triggered, repeat every 1 hour for a duration of 1 day.	7/27/2021 2:30:00 PM	7/27/2021 1:52:11 PM
SubmitAgent	Ready	At 9:00 AM every day - After triggered, repeat every 1 hour for a duration of 1 day.	7/27/2021 2:00:00 PM	7/27/2021 1:48:44 PM



EnrollAgent Properties (Local Computer)

General | Triggers | Actions | Conditions | Settings | History

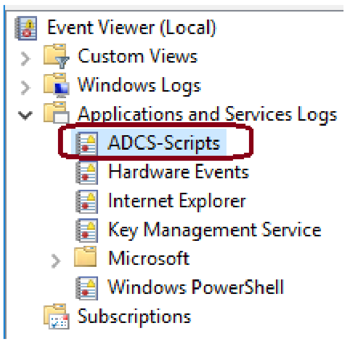
Name: EnrollAgent
Location: \Microsoft\ADCS
Author: FABRIKAM\installaccount
Description: Certificate enroll agent

Security options
When running the task, use the following user account:
ReqAGNTgMSA$ [Change User or Group...]
( ) Run only when user is logged on
(●) Run whether user is logged on or not
☐ Do not store password. The task will only have access to local computer resources.
☐ Run with highest privileges

☑ Hidden   Configure for: Windows Server 2016

[OK] [Cancel]

- Create a special event log to write status information into.



Event Viewer (Local)
- Custom Views
- Windows Logs
- Applications and Services Logs
  - ADCS-Scripts
  - Hardware Events
  - Internet Explorer
  - Key Management Service
  - Microsoft
  - Windows PowerShell
- Subscriptions

The installation routine will highlight additional manual tasks that need to be processed, such as working directory creation and structure and agent account permissions (see Introduction for more information).

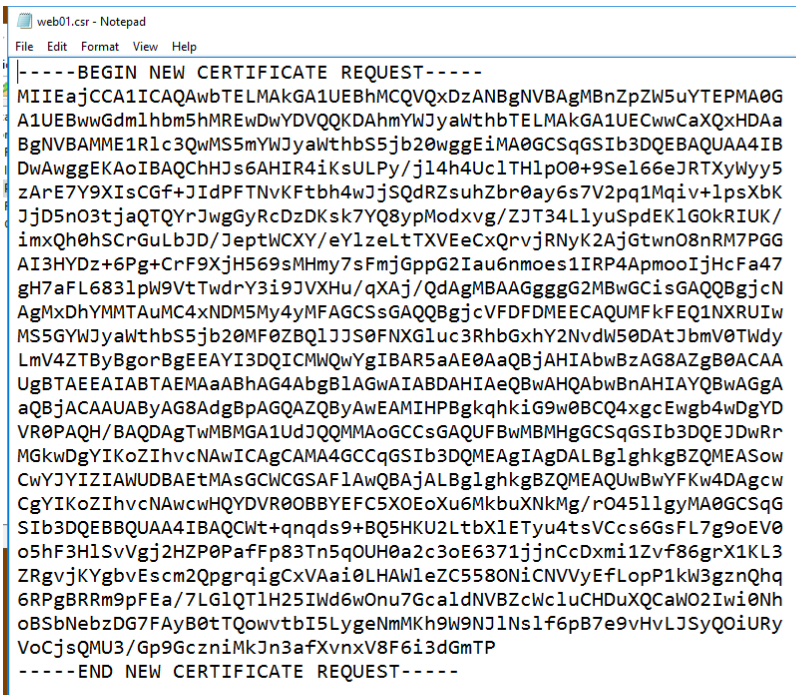
Request Processing Workflow

Incoming Requests

Incoming request can include three different parts:

1. The certificate request (mandatory)

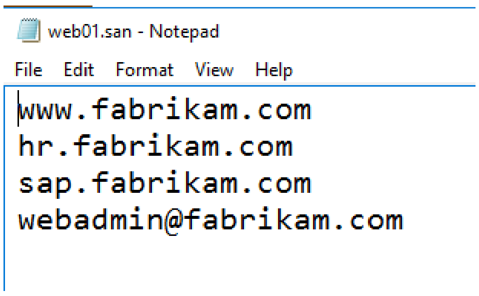
   The certificate request must come in standard base64 encoding file with the file extension "**.csr**"



```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIEajCCA1ICAQAwbTELMAkGA1UEBhMCQVQxDzANBgNVBAgMBnZpZW5uYTEPMA0G
A1UEBwwGdmlhbm5hMREwDwYDVQQKDAhmYWJyaWthbTELMAkGA1UECwwCaXQxHDAa
BgNVBAMME1Rlc3QwMS5mYWJyaWthbS5jb20wggEiMA0GCSqGSIb3DQEBAQUAA4IB
DwAwggEKAoIBAQChHJs6AHIR4iKsULPy/jl4h4UclTHlpO0+9Sel66eJRTXyWyy5
zArE7Y9XIsCGf+JIdPFTNvKFtbh4wJjSQdRZsuhZbr0ay6s7V2pq1Mqiv+lpsXbK
JjD5nO3tjaQTQYrJwgGyRcDzDKsk7YQ8ypModxvg/ZJT34LlyuSpdEKlGOkRIUK/
imxQh0hSCrGuLbJD/JeptWCXY/eYlzeLtTXVEeCxQrvjRNyK2AjGtwnO8nRM7PGG
AI3HYDz+6Pg+CrF9XjH569sMHmy7sFmjGppG2Iau6nmoes1IRP4ApmooIjHcFa47
gH7aFL683lpW9VtTwdrY3i9JVXHu/qXAj/QdAgMBAAGgggG2MBwGCisGAQQBgjcN
AgMxDhYMMTAuMC4xNDM5My4yMFAGCSsGAQQBgjcVFDFDMEECAQUMFkFEQ1NXRUIw
MS5GYWJyaWthbS5jb20MF0ZBQlJJS0FNXGluc3RhbGxhY2NvdW50DAtJbmV0TWdy
LmV4ZTByBgorBgEEAYI3DQICMWQwYgIBAR5aAE0AaQBjAHIAbwBzAG8AZgB0ACAA
UgBTAEEAIABTAEMAaABhAG4AbgBlAGwAIABDAHIAeQBwAHQAbwBnAHIAYQBwAGgA
aQBjACAAUAByAG8AdgBpAGQAZQByAwEAMIHPBgkqhkiG9w0BCQ4xgcEwgb4wDgYD
VR0PAQH/BAQDAgTwMBMGA1UdJQQMMAoGCCsGAQUFBwMBMHgGCSqGSIb3DQEJDwRr
MGkwDgYIKoZIhvcNAwICAgCAMA4GCCqGSIb3DQMEAgIAgDALBglghkgBZQMEASow
CwYJYIZIAWUDBAEtMAsGCWCGSAFlAwQBAjALBglghkgBZQMEAQUwBwYFKw4DAgcw
CgYIKoZIhvcNAwcwHQYDVR0OBBYEFC5XOEoXu6MkbuXNkMg/rO45llgyMA0GCSqG
SIb3DQEBBQUAA4IBAQCWt+qnqds9+BQ5HKU2LtbXlETyu4tsVCcs6GsFL7g9oEV0
o5hF3HlSvVgj2HZP0PafFp83Tn5qOUH0a2c3oE6371jjnCcDxmi1Zvf86grX1KL3
ZRgvjKYgbvEscm2QpgrqigCxVAai0LHAWleZC558ONiCNVVyEfLopP1kW3gznQhq
6RPgBRRm9pFEa/7LGlQTlH25IWd6wOnu7GcaldNVBZcWcluCHDuXQCaWO2Iwi0Nh
oBSbNebzDG7FAyB0tTQowvtbI5LygeNmMKh9W9NJlNslf6pB7e9vHvLJSyQOiURy
VoCjsQMU3/Gp9GczniMkJn3afXvnxV8F6i3dGmTP
-----END NEW CERTIFICATE REQUEST-----
```

2. The SAN extension file (optional)

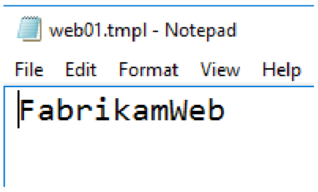
   In case, the presented request is missing SAN extensions, the requestor can provide a textfile containing additional SANs as simple text list. The file name prefix must be identical to the request file name prefix; the file extension must be "**.san**" in this case.



```
www.fabrikam.com
hr.fabrikam.com
sap.fabrikam.com
webadmin@fabrikam.com
```

3. The certificate template file (optional)

   If multiple certificate templates have been configured (and will be provided) for the usage together with the Automatic Request Processing Agents, the requestor can add a file which specify the certificate template to use. The specified template has to be allowed/configured in the config XML file (see section: Certificate Templates) and must have a file extension of "**.tmpl**". Again, the file name prefix must be identical to the request file name prefix- If no certificate template file has been provided, the configured default certificate template will be used (see section: Certificate Templates). The template file is in text format.

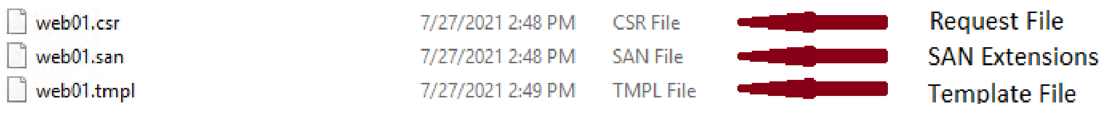


web01.tmpl - Notepad

File Edit Format View Help

```
FabrikamWeb
```

All three files must have the same filename prefix!



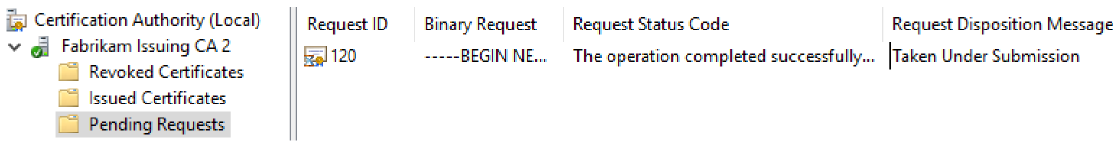
web01.csr	7/27/2021 2:48 PM	CSR File	Request File
web01.san	7/27/2021 2:48 PM	SAN File	SAN Extensions
web01.tmpl	7/27/2021 2:49 PM	TMPL File	Template File

Submitted Requests

After the incoming requests have been processed and submitted to the CA that has been configured in the config XML file, the requests are pending on that CA, waiting for Certificate Manager approval.

```
<!--
    Description:
        The CaName Node specifies the target CA for the request automation
        The name has to be in format "fqdn\CA name"
-->
<CaName>adcsca04.fabrikam.com\Fabrikam Issuing CA 2</CaName>
```

This is necessary as only in pending state, additional SANs can be added to the request.



Certification Authority (Local)
Fabrikam Issuing CA 2
Revoked Certificates
Issued Certificates
Pending Requests

Request ID	Binary Request	Request Status Code	Request Disposition Message
120	-----BEGIN NE...	The operation completed successfully...	Taken Under Submission

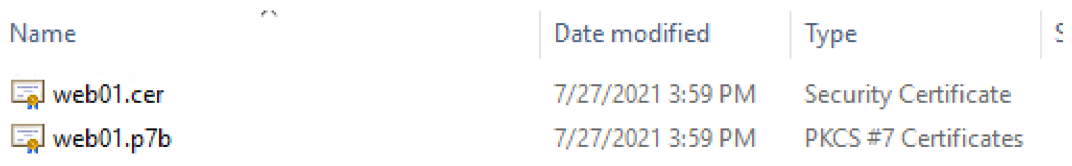
You can configure "AutoApproval" for specific templates in the config XML file. These templates must be part of the generally allowed/configured templates (see "***Certifcate Templates***" section earlier in this guide for more information).

```
<!--
    Description:
        The AutoApprove Node specifies whether auto-approval for pending requests based on
        specific named templates are allowed
        NOTE: It is highle recommended to NOT enable this functionality for security reasons
-->
<AutoApprove>
    <EnabledTemplate>FabrikamWeb3</EnabledTemplate>
</AutoApprove>
```

When “auto approval” is configured, the enrollment agents will automatically approve requests based on these templates.

> **Note:** We strongly recommend to NOT enable this functionality. The Certificate Manager is ultimately responsible for compliance of the issued certificates. Enabling “auto approval” functionality violates this role definition!

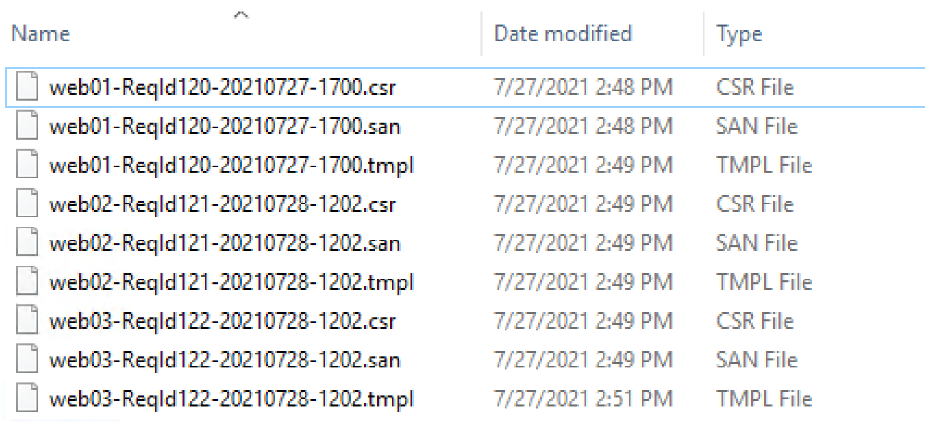
When the request has been approved by a Certificate Manager, the workflow continues to enroll the appropriate certificates into the configured “outbox” folder in both, binary (DER) and P7B format. The enrolled files continue to have the same file name prefex as the incoming request file.



Name	Date modified	Type
web01.cer	7/27/2021 3:59 PM	Security Certificate
web01.p7b	7/27/2021 3:59 PM	PKCS #7 Certificates

Archiving the Requests

After the incoming request have been processed, they will get moved into the configured archive folder with the request ID and an added timestamp of the processing.



Name	Date modified	Type
web01-ReqId120-20210727-1700.csr	7/27/2021 2:48 PM	CSR File
web01-ReqId120-20210727-1700.san	7/27/2021 2:48 PM	SAN File
web01-ReqId120-20210727-1700.tmpl	7/27/2021 2:49 PM	TMPL File
web02-ReqId121-20210728-1202.csr	7/27/2021 2:49 PM	CSR File
web02-ReqId121-20210728-1202.san	7/27/2021 2:49 PM	SAN File
web02-ReqId121-20210728-1202.tmpl	7/27/2021 2:49 PM	TMPL File
web03-ReqId122-20210728-1202.csr	7/27/2021 2:49 PM	CSR File
web03-ReqId122-20210728-1202.san	7/27/2021 2:49 PM	SAN File
web03-ReqId122-20210728-1202.tmpl	7/27/2021 2:51 PM	TMPL File

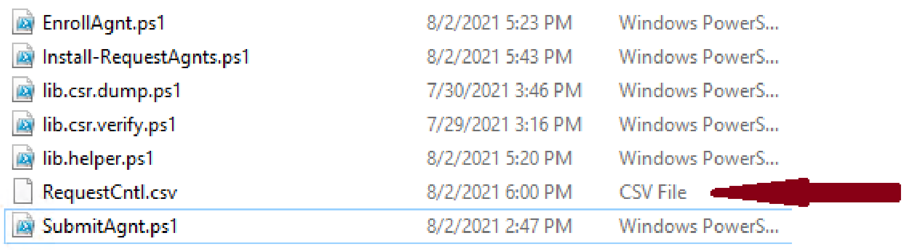
Request Clean Up

You can define a “CleanUp” duration that will be used to delete old entries from the configured outbox and archive folders that are older than the defined CleanUp duration.

```xml
<!--
    Description:
        The CleanUpDuration Node specifies time in days after that the files in folder will be deleted
-->
<CleanUpDuration>14</CleanUpDuration>
```

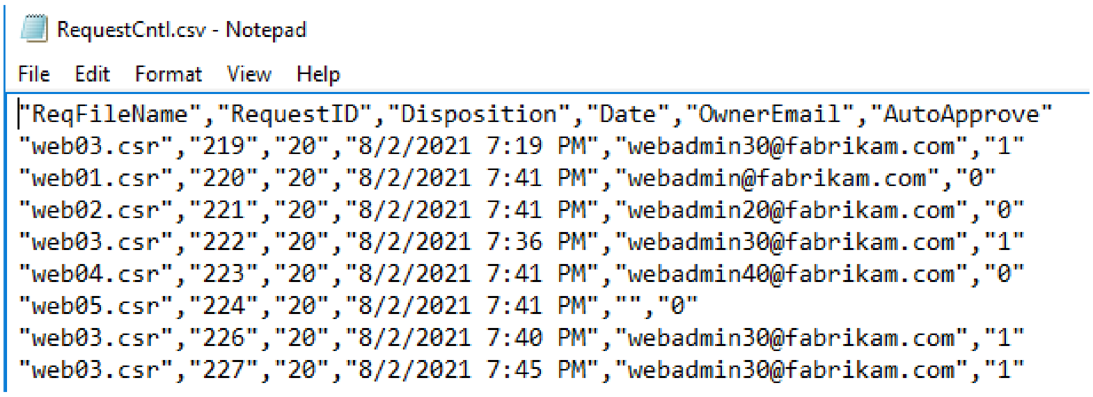
The Control File

In the startup folder of the agent scripts, a control file will be created to track the status of the incoming requests. This file will never be automatically cleaned up and can be used for long term tracking of incoming requests.



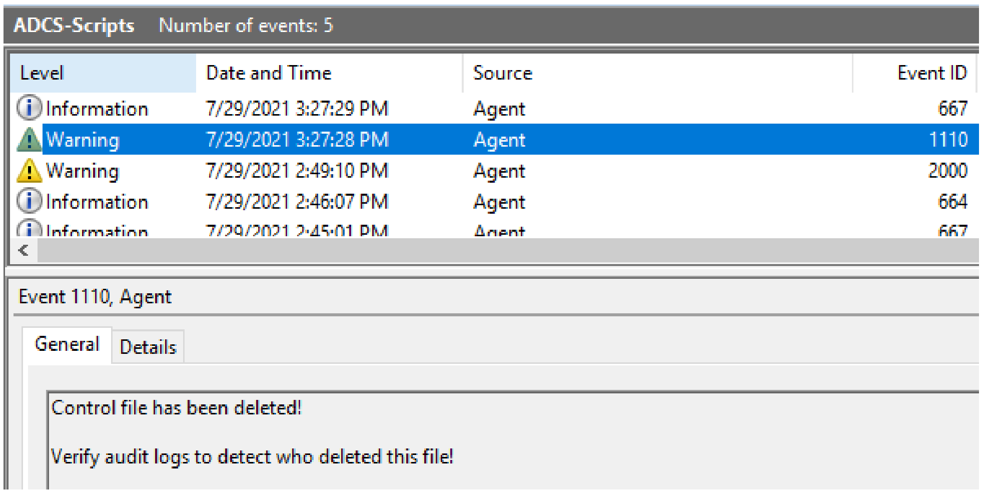
```
EnrollAgnt.ps1              8/2/2021 5:23 PM    Windows PowerS...
Install-RequestAgnts.ps1    8/2/2021 5:43 PM    Windows PowerS...
lib.csr.dump.ps1            7/30/2021 3:46 PM   Windows PowerS...
lib.csr.verify.ps1          7/29/2021 3:16 PM   Windows PowerS...
lib.helper.ps1              8/2/2021 5:20 PM    Windows PowerS...
RequestCntl.csv             8/2/2021 6:00 PM    CSV File
SubmitAgnt.ps1              8/2/2021 2:47 PM    Windows PowerS...
```

The control file is in CSV format and contains the original request file name, the request ID at the CA database, the time stamp when the request has been submitted to the CA and the request owner email if one has been defined in the SAN file.



```
"ReqFileName","RequestID","Disposition","Date","OwnerEmail","AutoApprove"
"web03.csr","219","20","8/2/2021 7:19 PM","webadmin30@fabrikam.com","1"
"web01.csr","220","20","8/2/2021 7:41 PM","webadmin@fabrikam.com","0"
"web02.csr","221","20","8/2/2021 7:41 PM","webadmin20@fabrikam.com","0"
"web03.csr","222","20","8/2/2021 7:36 PM","webadmin30@fabrikam.com","1"
"web04.csr","223","20","8/2/2021 7:41 PM","webadmin40@fabrikam.com","0"
"web05.csr","224","20","8/2/2021 7:41 PM","","0"
"web03.csr","226","20","8/2/2021 7:40 PM","webadmin30@fabrikam.com","1"
"web03.csr","227","20","8/2/2021 7:45 PM","webadmin30@fabrikam.com","1"
```

The control file is hashed evey time when it is written to the file system and this hash is verified when it is accessed. So, manual modifications or even file deletion is tracked and is reported.



Level	Date and Time	Source	Event ID
Information	7/29/2021 3:27:29 PM	Agent	667
Warning	7/29/2021 3:27:28 PM	Agent	1110
Warning	7/29/2021 2:49:10 PM	Agent	2000
Information	7/29/2021 2:46:07 PM	Agent	664
Information	7/29/2021 2:45:01 PM	Agent	667

Control file has been deleted!

Verify audit logs to detect who deleted this file!

Process Status Information

"The Automatic Request Processing workflow generates status information in the event log or via email. Administrative events will be written in any case into the defined eventlog. See Appendix for events and event IDs.

Status Information on local Event Log

However, to write enrollment status events into the defined eventlog, the config XML file needs to be prepared accordingly.

```
<Eventlog>
    <!-- write success events to the eventlog (true/false) -->
    <UseEventlog>True</UseEventlog>
```

Additionally, you can define, if you only want to generate events for occurring enrollment issues or if you want to get any status information.

```
<!-- write success events to the eventlog (true/false) -->
<WriteSuccessEvents>True</WriteSuccessEvents>
```

You can as well define the event log to be used, the event source and the event ids that should be used for the different log entries.

```
<!-- define the event log that should be used for events -->
<EventLog>ADCS-Scripts</EventLog>

<!-- define the event source that should be used for events in the eventlog -->
<EventSource>ADCS-Enroll-Agent</EventSource>

<!-- define the event ID that should be used for failure events in the eventlog -->
<FailEventID>666</FailEventID>

<!-- define the event ID that should be used for submit events in the eventlog -->
<SubmitEventID>667</SubmitEventID>

<!-- define the event ID that should be used for issue events in the eventlog -->
<IssueEventID>665</IssueEventID>

<!-- define the event ID that should be used for enroll events in the eventlog -->
<EnrollEventID>664</EnrollEventID>
```

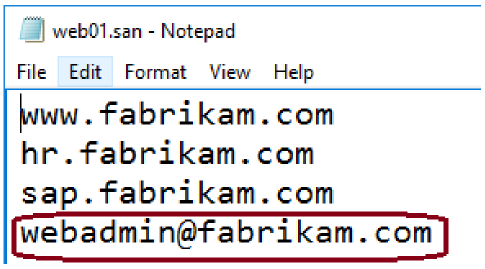
Independent messages can also be specified for each enrollment status of the workflow.

```
<!-- define the event msg for submit request events in the eventlog -->
<SubmitEventMsg>The request !REQName!, RequestID: !REQID! has been submitted and is waiting for formal approval.</SubmitEventMsg>

<!-- define the event msg for enroll events in the eventlog -->
<EnrollEventMsg>The request !REQName!, RequestID: !REQID! has been enrolled.</EnrollEventMsg>

<!-- define the event msg for enroll events in the eventlog -->
<FailedEventMsg>The request !REQName!, RequestID: !REQID! has been failed.</FailedEventMsg>

<!-- define the event msg for enroll events in the eventlog -->
<DeniedEventMsg>The request !REQName!, RequestID: !REQID! has been denied.</DeniedEventMsg>
```

Note: Terms enveloped by "!" are place holders and will be replaced during run time.

Status Information via Email

Email-based status information will only be sent, if the request associated SAN file does contain an email address-based SAN (RFC822-based SAN).



```
www.fabrikam.com
hr.fabrikam.com
sap.fabrikam.com
webadmin@fabrikam.com
```

In such case, the email address will get interpreted as the requestor's (or requestors group) email address and will be used to automatically send information about the current status of the process. In this case, it is recommended to define the "Certificate Managers" or the "PKI Administrators" email address as CC address to have them informed about all communications.

To make email status information work, a couple of configuration items must be filled in the config XML file.

```
<Mail>
    <Server>adcsdc01.fabrikam.com</Server>
    <Port>25</Port>
    <UseSSL>false</UseSSL>
    <SSLPort>25</SSLPort>
    <MailFrom>CertificateAutomation@fabrikam.com</MailFrom>
    <MailCC>PKIAdmins@fabrikam.com</MailCC>
    <Subject>Incoming Certificate Request Status</Subject>
```

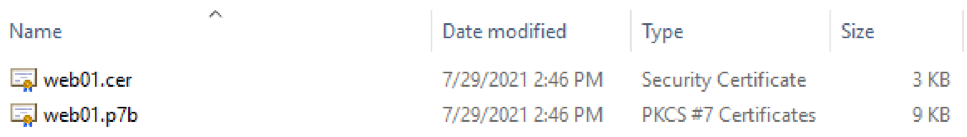
Additionally, individual messages can be defined for the different enrollment process states.

```
        <SubmitMsg>
Hello,

this is to inform you that your request !REQName! has been submitted and is waiting for formal approval.
You will get informed if the status will change!

Thank you!
        </SubmitMsg>

        <!--
            Message when successfully issued and enrolled a certificate
            words enveloped by "!" are placeholders and will get replaced during
            script run
        -->
        <EnrollMsg>
Hello,

your request !REQName! has been processed. You can download your certifcates using the following link:
file://!CERName!
file://!P7BName!

Thank you!
        </EnrollMsg>
```

Note: Again, terms enveloped by "!" are place holders and will be replaced during run time.

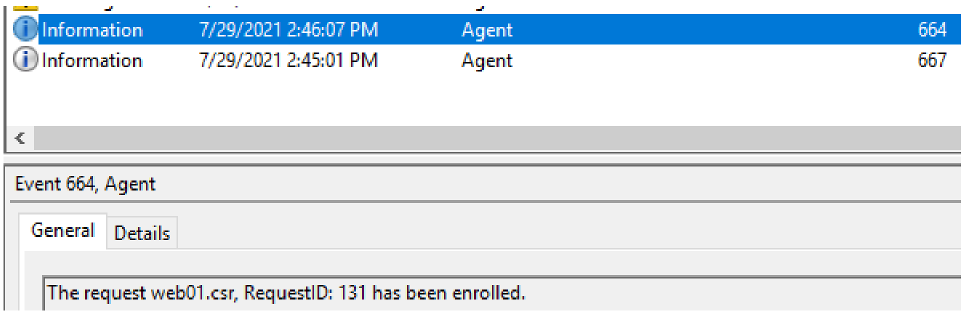
Retrieving the Enrolled Certificates

Finally, the certificates will made available for retrieval in the configured outbox folder.



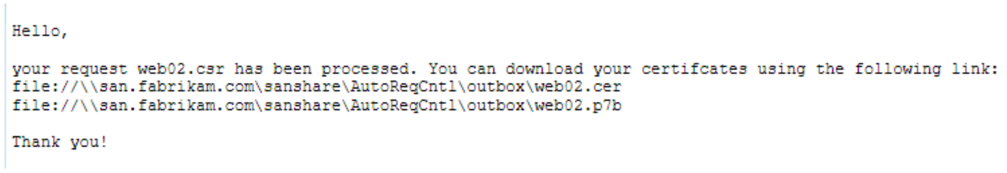
Name	Date modified	Type	Size
web01.cer	7/29/2021 2:46 PM	Security Certificate	3 KB
web01.p7b	7/29/2021 2:46 PM	PKCS #7 Certificates	9 KB

If event log status information has been enabled, an eventlog entry will get generated.



```
Information   7/29/2021 2:46:07 PM   Agent   664
Information   7/29/2021 2:45:01 PM   Agent   667

Event 664, Agent
General  Details
The request web01.csr, RequestID: 131 has been enrolled.
```

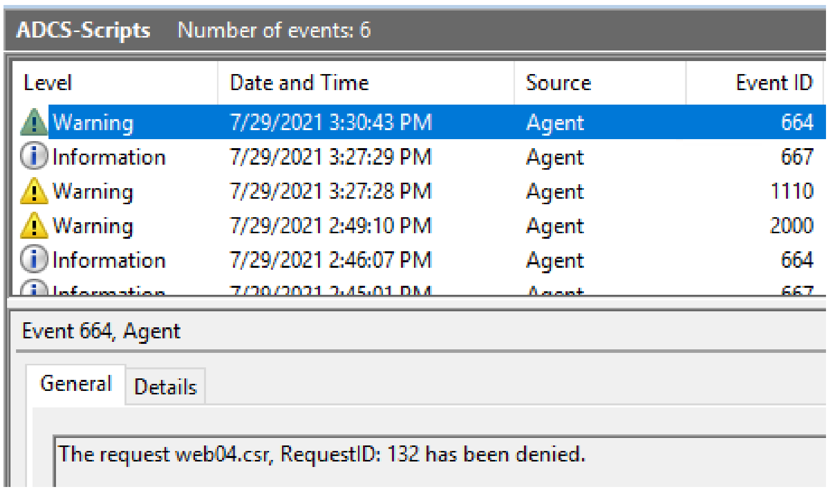
If an email-based SAN (RFC822) has been found in the SAN file, an email will be sent out to this email address with the configured "admin" email address on CC to inform where to fetch the enrolled certificate.



```
Hello,

your request web02.csr has been processed. You can download your certifcates using the following link:
file://\\san.fabrikam.com\sanshare\AutoReqCntl\outbox\web02.cer
file://\\san.fabrikam.com\sanshare\AutoReqCntl\outbox\web02.p7b

Thank you!
```

Denied or Failed Requests

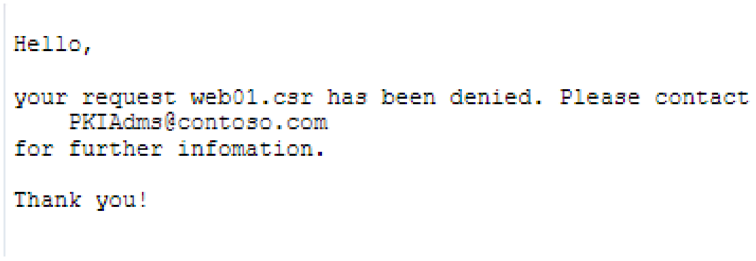
If a request has been denied by the approver, an eventlog entry will get generated if event log status information has been enabled. The same occurs when a request has been approved but failed the enrollment.



ADCS-Scripts   Number of events: 6

```
Level         Date and Time           Source   Event ID
Warning       7/29/2021 3:30:43 PM    Agent    664
Information   7/29/2021 3:27:29 PM    Agent    667
Warning       7/29/2021 3:27:28 PM    Agent    1110
Warning       7/29/2021 2:49:10 PM    Agent    2000
Information   7/29/2021 2:46:07 PM    Agent    664
Information   7/29/2021 2:45:01 PM    Agent    667

Event 664, Agent
General  Details
The request web04.csr, RequestID: 132 has been denied.
```

Again, if an email-based SAN has been found in the SAN file, an email will be sent out to this email address with the configured “admin” email address on CC.



```
Hello,

your request web01.csr has been denied. Please contact
    PKIAdms@contoso.com
for further infomation.

Thank you!
```

Uninstalling the Agents

To remove the scheduled tasks and any configuration artefact from the system, run:

```
Install-RequestAgnts.ps1 -UninstallAgents
```

The eventlog and existing events will not be touched. Also, the working directory and all containing data will stay and must be removed manually.

Updating the Agents Configuration

The enrollment agent’s running configuration is stored in a special registry hive:

```
HKLM:\SYSTEM\CurrentControlSet\Services\CertSvc\EnrollAgents
```

It is strongly recommended that no modifications directly on the registry entries are performed. For all configuration changes modify the **config.xml** file and run:

```
Install-RequestAgnts.ps1 -UpdateConfig
```

This will update the running configuration for the agents.

The following configuration items can only be modified by uninstalling and reinstalling the enrollment agents:

- AgentAccountName
- SubmitTaskName
- EnrollTaskName
- TaskRepetitionInterval

When changing the following configuration items:

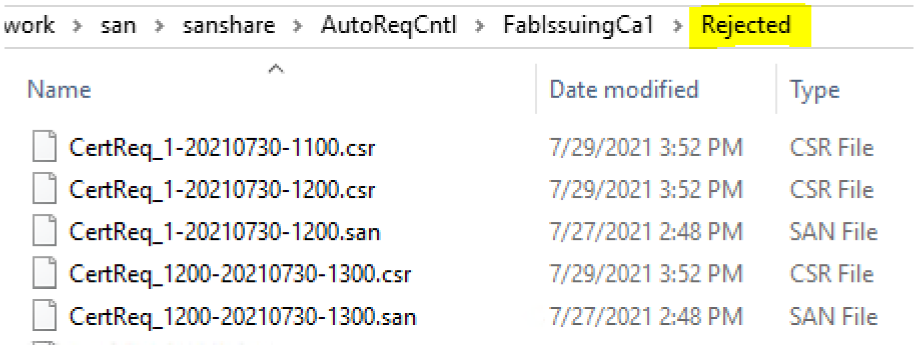
- Eventlog
- EventSource.

the old eventlog and event source will stay available and must be removed manually if not needed anymore.

Pre-validating incoming requests

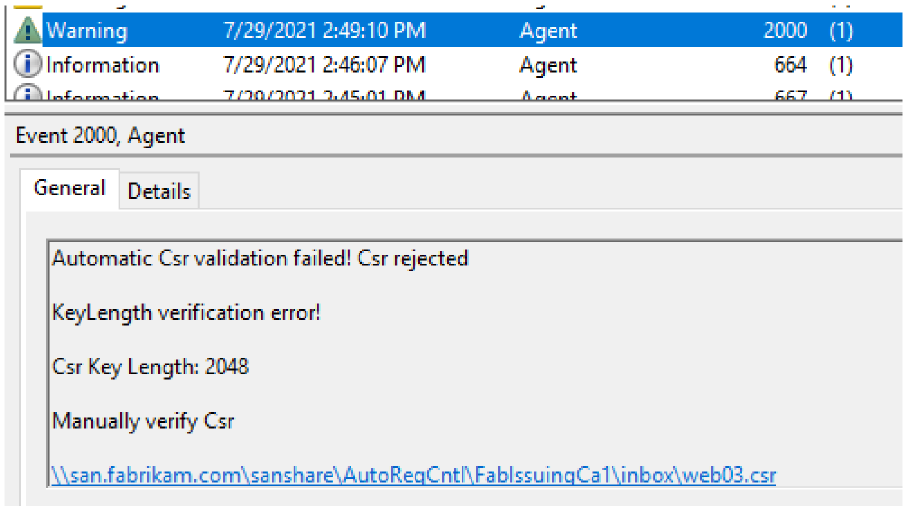
Before an incoming request will be processed, a pre-validation check can be enfored. The pre-validation check consists of several specially defined verification steps, ensuring that the request does not violate the compliance standards.

All verification steps must be wrapped into PowerShell functions and included in the lib.csr.verify.ps1 file. The pre-validation function (`Pass-CsrVerification`) is returning the result to the agent's main function. Requests failing the pre-validation are moved into the "rejected" folder for further investigation.


work › san › sanshare › AutoReqCntl › FabIssuingCa1 › Rejected

Name	Date modified	Type
CertReq_1-20210730-1100.csr	7/29/2021 3:52 PM	CSR File
CertReq_1-20210730-1200.csr	7/29/2021 3:52 PM	CSR File
CertReq_1-20210730-1200.san	7/27/2021 2:48 PM	SAN File
CertReq_1200-20210730-1300.csr	7/29/2021 3:52 PM	CSR File
CertReq_1200-20210730-1300.san	7/27/2021 2:48 PM	SAN File

In addition, an administrative event is generated.



Warning	7/29/2021 2:49:10 PM	Agent	2000	(1)
Information	7/29/2021 2:46:07 PM	Agent	664	(1)

```
Automatic Csr validation failed! Csr rejected

KeyLength verification error!

Csr Key Length: 2048

Manually verify Csr

\\san.fabrikam.com\sanshare\AutoReqCntl\FabIssuingCa1\inbox\web03.csr
```

Custom verification steps can be defined based on the following Csr items: (see ***Appendix*** for more information)

- `SignatureMatch`
- `RequestAttributes`
- `Extensions`
- `SubjectName`
- `clientInfo`
- `CertTmpl`
- `EnrollmentCSP`
- `EnhancedKeyUsage`
- `KeyUsage`
- `SAN`
- `signAlgo`
- `KeyAlgo`
- `KeyLength`

Appendix

Arguments accepted by the installation script (Install-RequestAgnts.ps1)

Argument	Required	Description
InstallAgents	optional	Define if the agents will be installed as scheduled task; configuration items are taken from the config.xml file Default value: **false**.
UninstallAgents	optional	Remove the agent's scheduled tasks and all connected registry values Default value: **false**.
UpdateConfig	optional	Updates agent's configuration in registry from config XML file Default value: **false**.

Table 1: Arguments accepted by the Installation Script

Fixed event IDs of the agents

EventID	Description
1000	Active Directory Certificate Services is not running.
1001	Configured working directory is not accessible.
1002	Configured "inbox" folder is not accessible.
1003	Configured "outbox" folder is not accessible.
1004	Configured "archive" folder is not accessible.
1005	Configured "rejected" folder is not accessible.
1010	Move-Item failed! During request submission, the processed request will be moved into either the archive or rejected folder. This failed for the mentioned request files.
1020	Request not found in CA db
1050	Auto Approval error
1055	Auto Approval succeeded
1100	CleanUp failed. During the enrollment, the agents will clean up the output and archive folders based on the CleanUp configuration entry. This failed with the described error.
1110	Audit Failure. File hash of control file failed hash comparism. The file might have been modified offline.
1111	General Send-Mail failure.
2000	Automatic Csr validation error. Csr was rejected

Table 2: Arguments accepted by the Installation Script

CSR structure for pre-validation

```
    TypeName: Selected.System.String

Name                 MemberType   Definition
----                 ----------   ----------
Equals               Method       bool Equals(System.Object obj)
GetHashCode          Method       int GetHashCode()
GetType              Method       type GetType()
ToString             Method       string ToString()
CertTmpl             NoteProperty Object[] CertTmpl=System.Object[]
ClientInfo           NoteProperty object ClientInfo=null
EnhancedKeyUsage     NoteProperty Selected.System.String EnhancedKeyUsage=@{Name=Server
Authentication ; OID=1.3.6.1.5.5.7.3.1}...
EnrollmentCSP        NoteProperty string EnrollmentCSP= Microsoft RSA SChannel Cryptographic Provider
Extensions           NoteProperty Object[] Extensions=System.Object[]
KeyAlgo              NoteProperty Selected.System.String KeyAlgo=@{AlgoOID=1.2.840.113549.1.1.1;
AlgoName=RSA}
KeyLength            NoteProperty int KeyLength=2048
KeyUsage             NoteProperty string KeyUsage=Digital Signature, Key Encipherment (a0)...
RequestAttributes    NoteProperty Object[] RequestAttributes=System.Object[]
SAN                  NoteProperty Object[] SAN=System.Object[]
SignAlgo             NoteProperty Selected.System.String SignAlgo=@{AlgoOID=1.2.840.113549.1.1.5;
AlgoName=sha1RSA}
SignatureMatch       NoteProperty bool SignatureMatch=True
SubjectName          NoteProperty Selected.System.String SubjectName=@{CN= hrweb01.fabrikam.com;
OU=HR-Ger; OHR; L=FFM; ST=; STREET=Dumm> Srteet; C=DE; E=Webadmin01@fabrikam.com}
```

Subject name

```
    TypeName: Selected.System.String

Name        MemberType   Definition
----        ----------   ----------
Equals      Method       bool Equals(System.Object obj)
GetHashCode Method       int GetHashCode()
GetType     Method       type GetType()
ToString    Method       string ToString()
C           NoteProperty string C=DE
CN          NoteProperty string CN=hrweb01.fabrikam.com
E           NoteProperty string E=Webadmin01@fabrikam.com
L           NoteProperty object L=FFM
O           NoteProperty string O=HR
OU          NoteProperty string OU=HR-Ger
S           NoteProperty object S=Hessen
STREET      NoteProperty object STREET=Irgendwostrasse
```

Example:

```
$Csr.SubjectName
```

```
CN     : web01.fabrikam.com
OU     : HR-DE
O      : HR
L      : FFM
S      : Hessen
STREET : IrgendwoStrasse
C      : DE
E      : webadmin01.fabrikam.com
```

Subject alternative names

```
    TypeName: Selected.System.String

Name        MemberType   Definition
----        ----------   ----------
Equals      Method       bool Equals(System.Object obj)
GetHashCode Method       int GetHashCode()
GetType     Method       type GetType()
ToString    Method       string ToString()
SAN         NoteProperty string SAN=san1.fabrikam.de
Type        NoteProperty string Type=DNS Name
```

Example:

```
$Csr.SAN
```

```
Type     SAN
----     ---
DNS Name san1.fabrikam.de
DNS Name san2.fabrikam.de
DNS Name san3.fabrikam.de
DNS Name san4.fabrikam.de
DNS Name san5.fabrikam.de
DNS Name san6.fabrikam.de
DNS Name www.fabrikam.com
DNS Name hr.fabrikam.com
DNS Name sap.fabrikam.com
Rcf822   webadmin@fabrikam.com
```

Enhanced Key Usages (EKU)

```
    TypeName: Selected.System.String

Name        MemberType   Definition
----        ----------   ----------
Equals      Method       bool Equals(System.Object obj)
GetHashCode Method       int GetHashCode()
GetType     Method       type GetType()
ToString    Method       string ToString()
Name        NoteProperty string Name=Server Authentication
OID         NoteProperty string OID=1.3.6.1.5.5.7.3.1
```

Example:

```
$Csr.EnhancedKeyUsage
```

```
Name                     OID
----                     ---
Server Authentication    1.3.6.1.5.5.7.3.1
Client Authentication    1.3.6.1.5.5.7.3.2
```