

# Certificate Import Tool



## Administration manual for system providers

5/31/2022

### Product line Neo, version 7

The described functions can be used with the following ASC products:

EVOIP<sup>neo</sup>

EVOLUTION<sup>neo</sup> / XXL / eco

Please note that you can always find the most up-to-date technical documentation and product updates in the partner area on our website at <https://www.asctechnologies.com>.

Copyright © 2022 ASC Technologies AG. All rights reserved.

Windows is a registered trademark of Microsoft Corporation. VMware® is a registered trademark of VMware, Inc. All other marks and names mentioned herein may be trademarks of their respective companies.

**ASC Technologies AG - Seibelstr. 2-4 - 63768 Hösbach - Germany**



## Contents

<b>1</b>	<b>General information.....</b>	<b>4</b>
<b>2</b>	<b>Introduction.....</b>	<b>5</b>
<b>3</b>	<b>HTTPS certificates .....</b>	<b>6</b>
3.1	HTTPS Certificate.....	6
3.1.1	Tab About .....	6
3.1.2	Tab Current Certificate .....	6
3.1.3	Tab Import Certificate .....	7
3.2	HTTPS Trust.....	8
3.2.1	Tab About .....	8
3.2.2	Tab Current Trusted .....	9
3.2.3	Tab Import Trusted Certificate.....	10
<b>4</b>	<b>PBX certificates .....</b>	<b>11</b>
4.1	PBX Certificate .....	11
4.1.1	Tab About .....	11
4.1.2	Tab Current Certificate .....	11
4.1.3	Tab Import Certificate .....	12
4.2	PBX Trust .....	13
4.2.1	Tab About .....	13
4.2.2	Tab Current Trusted .....	14
4.2.3	Tab Import Trusted Certificate.....	14
<b>5</b>	<b>Generate Certificate.....</b>	<b>17</b>
5.1	Generic Certificate Import.....	17
5.2	Generic Trust Import.....	17
<b>6</b>	<b>Generate Request .....</b>	<b>19</b>
6.1	Generate Certificate.....	19
6.2	Generate CSR .....	20
<b>7</b>	<b>Reset all with self signed .....</b>	<b>21</b>
<b>8</b>	<b>Call up Certimporter in command line.....</b>	<b>22</b>
	<b>List of figures .....</b>	<b>24</b>
	<b>List of tables.....</b>	<b>25</b>
	<b>Glossary .....</b>	<b>26</b>

**General information**

In the context of this document ASC represents ASC Technologies AG, its subsidiaries, branch offices, and distributors. An up-to-date overview of the aforementioned entities can be found at <https://www.asctechnologies.com>

ASC assumes no guarantee for the actuality, correctness, integrity or quality of the information provided in the manuals.

ASC regularly checks the content of the released manuals for consistency with the described hardware and software. Nevertheless, deviations cannot be excluded. Necessary revisions are included in subsequent editions.

Some aspects of the ASC technology are described in general terms to protect the ownership and the confidential information or trade secrets of ASC.

The software programs and the manuals of ASC are protected by copyright law. All rights on the manuals are reserved including the rights of reproduction and multiplication of any kind, be it photo mechanical, typographical or on digital data media. This also applies to translations. Copying the manuals, completely or in parts, is only allowed with written authorization of ASC.

Representative, if not defined otherwise, is the technical status at the time of the delivery of the software, the devices and the manuals of ASC. Technical changes without specified announcements are reserved. Previous manuals lose their validity.

The general conditions of sales and delivery of ASC in their latest version apply.

The Certificate Import Tool is a standard component of the Neo software.

It serves to create certificates, to request certificates from other certification authorities, and to import certificates.

1. On the recording server, open the directory  
C:\Program Files (x86)\ASC\ASC Product Suite\scripts.
2. Start the file *certimporter.exe* as administrator.  
⇒ The window Certificate Import Tool appears.

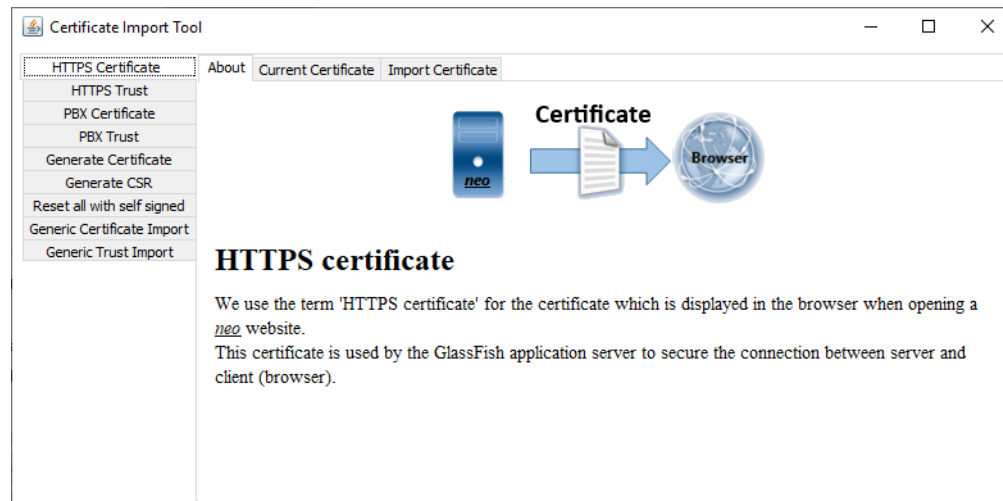


Fig. 1: Start Certificate Import Tool

### 3 HTTPS certificates

#### 3.1 HTTPS Certificate

This section holds the proprietary certificate that the web server displays when calling up a website for instance.

There are 3 tabs:

- *About*  
Information about the certificate, display, and usage,  
see [chapter "Tab About", p. 6](#).
- *Current Certificate*  
Information such as version, issuer, validity, IP addresses, DNS, algorithms,  
see [chapter "Tab Current Certificate", p. 6](#)
- *Import Certificate*  
Output keystore path, format selection, path of the certificate,  
see [chapter "Tab Import Certificate", p. 7](#).

##### 3.1.1 Tab About

The tab explains the purpose of the certificate.

We use the term *HTTPS Certificate* for the certificate which is displayed when opening a Neo website in the browser. This certificate is deployed by the GlassFish application server to protect the connection between the server and the client (browser).

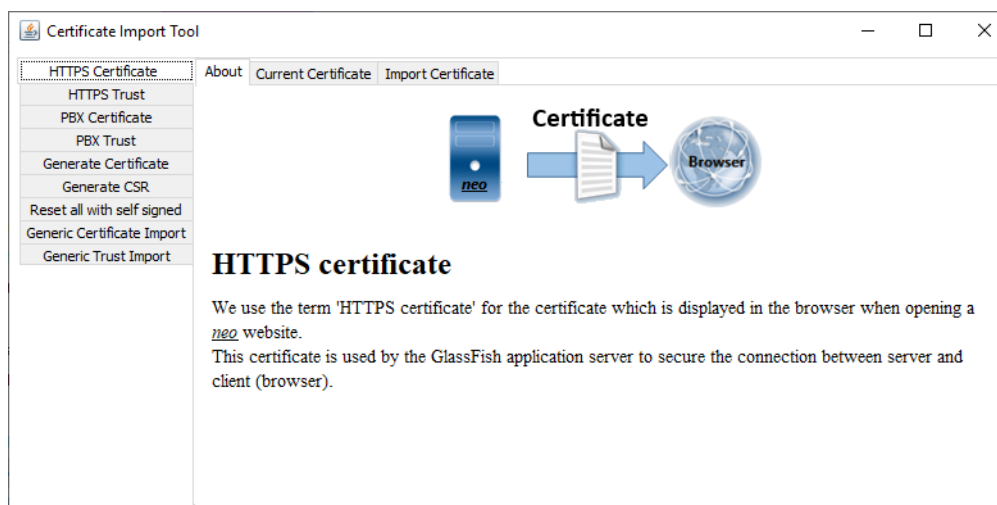


Fig. 2: HTTPS Certificate - tab About

##### 3.1.2 Tab Current Certificate

This tab displays information about the [HTTPS](#) certificate. This is the certificate that the recording server deploys to authenticate with a client, e. g. a browser.

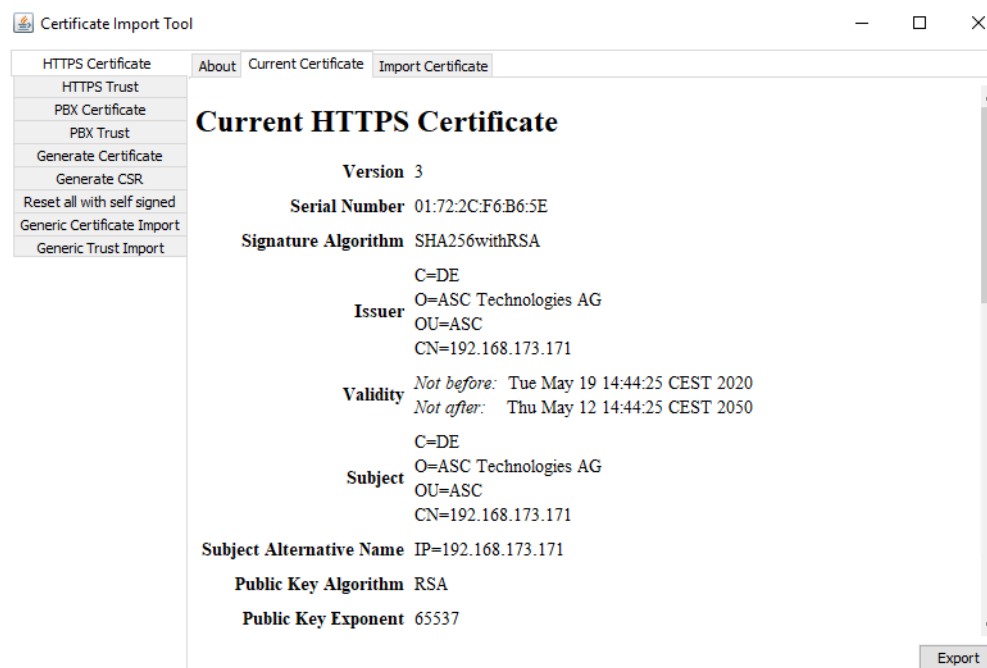


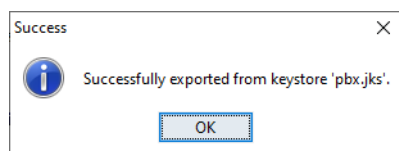
Fig. 3: Information about the current certificate

The following information is available:

- *Version*
- *Serial Number*
- *Signature Algorithm*
- *Issuer*
- *Validity*
- *Server details*
- *Alternative server name or DNS*
- *Public Key Algorithm*
- *Public Key Exponent*
- *Public Key*

By clicking on the button *Export*, you can export the certificate.

1. Click on the button *Export* to export the current self-signed certificate.
2. Select an appropriate storage location for the certificate.
3. Click on the button *Save*.
  - ⇒ A success message appears.



### 3.1.3 Tab Import Certificate

In this tab, you can import a [HTTPS](#) certificate.

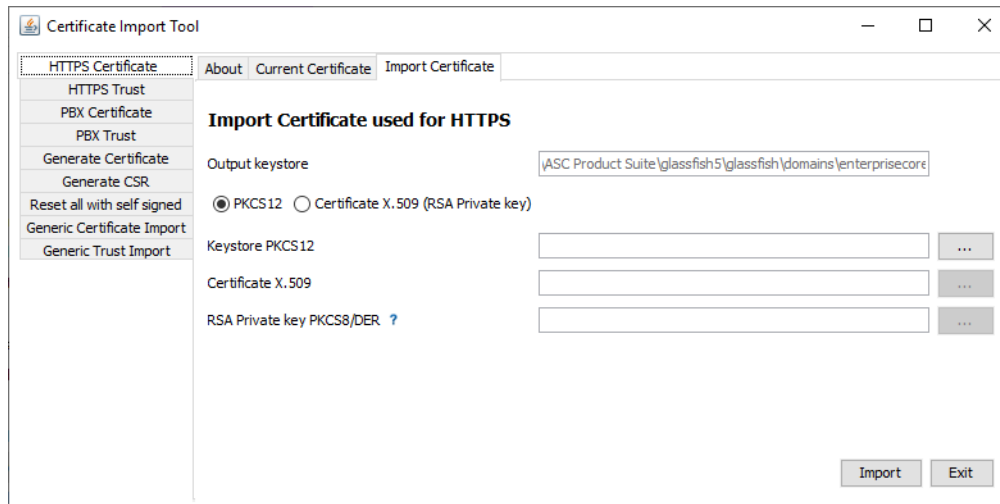



Fig. 4: HTTPS Certificate - Tab Import Certificate

The following formats are supported:

- **PKCS12**  
is a keystore format that contains the public key and the private key in a file.
  - **X.509**  
is the file format of a certificate that contains the public key. If you have requested the X.509 **RSA** certificate via a **CSR**, you must not complete the field **RSA Private Key** as the private key already exists on the system.  
If you do not have requested the X.509 **RSA** certificate via a **CSR**, you must import the private key separately by means of the field **RSA Private Key**.
1. The field *Output keystore* contains the path to the directory where external applications can find the certificates.
  2. Select the type of the certificate by activating the radio button.
  3. Click on the button  next to the respective format to select the file from the Explorer path.
  4. Click on the button *Import*.
- ⇒ A success message appears.

### 3.2 HTTPS Trust

This section displays the trusted root certificates, also called **CA** (certification authorities).

There are 3 tabs:

- **About**  
Information about the certificate, display, and usage,  
see [chapter "Tab About", p. 8](#).
- **Current Trusted**  
Information such as version, issuer, validity, IP addresses, DNS, algorithms,  
see [chapter "Tab Current Trusted", p. 9](#)
- **Import Trusted Certificate**  
Output keystore path, format selection, path of the certificate,  
see [chapter "Tab Import Trusted Certificate", p. 10](#).

#### 3.2.1 Tab About

The tab explains the purpose of the certificate.

We use the term *HTTPS Certificate* for certificates that the GlassFish application server trusts. This only become relevant in rare cases when the application server has to establish a connection to another web service or when using **LDAP**.



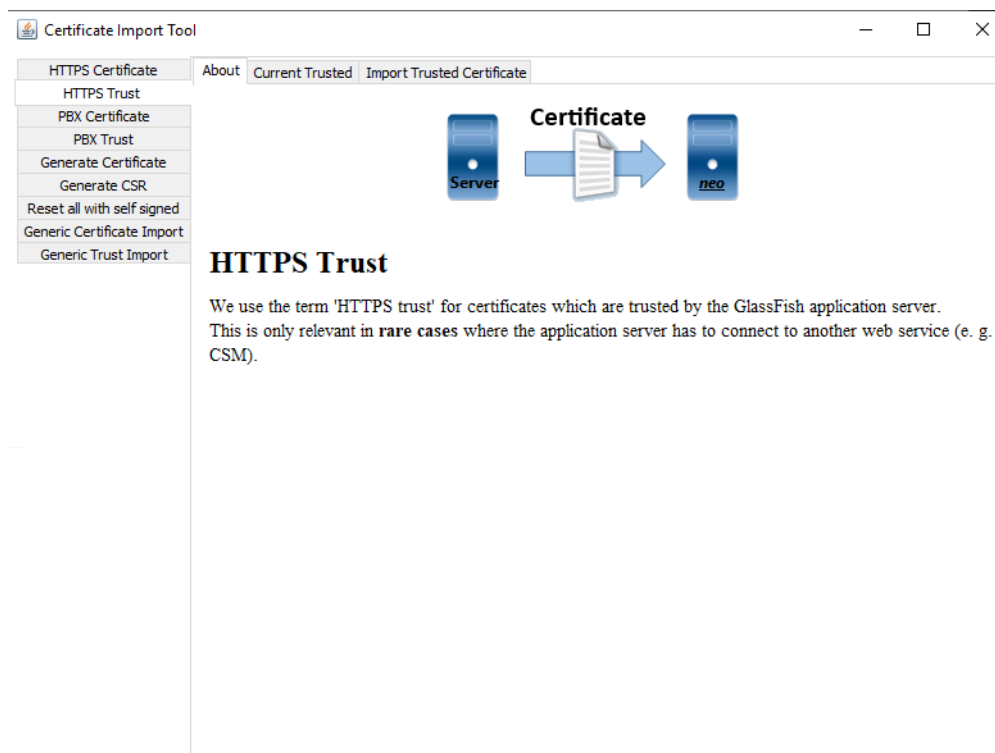


Fig. 5: HTTPS Trust - tab About

### 3.2.2 Tab Current Trusted

This tab displays information about the HTTPS Trust certificates.

All [CA](#) certificates available in the [Truststore](#) that the recording server may trust are listed.

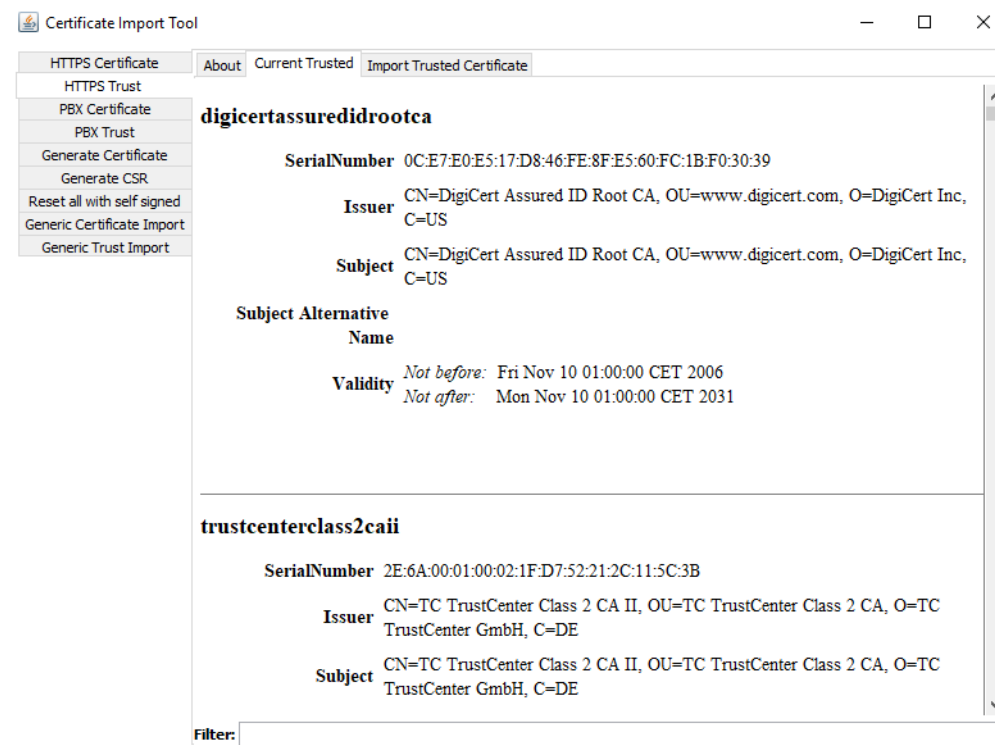


Fig. 6: HTTPS Trust Certificate - tab Current Trusted

Click on the context menu to delete individual certificates.

### 3.2.3 Tab Import Trusted Certificate

In this tab, you can import a [HTTPS](#) Trust certificate.

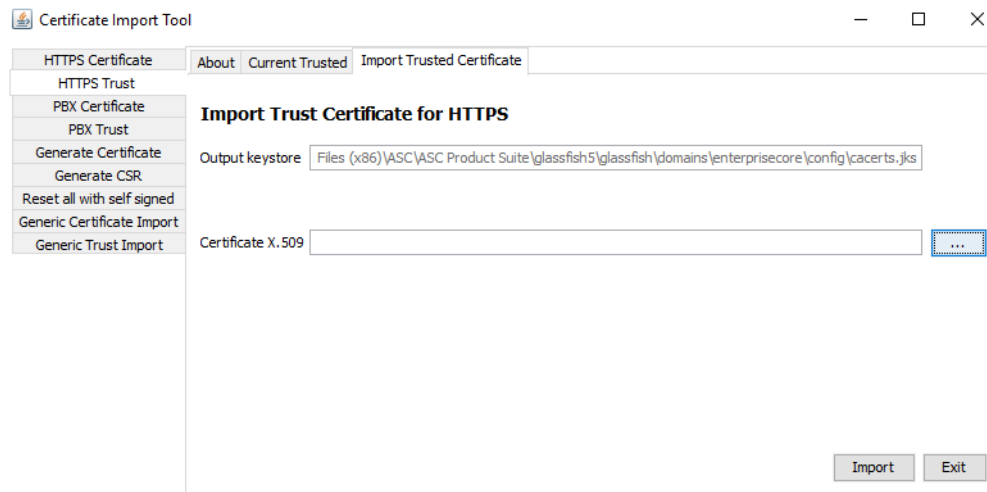


Fig. 7: HTTPS Trust - tab Import Trusted Certificate

1. The field *Output keystore* contains the path to the directory where external applications can find the certificates.
  2. Click on the button ... next to the field *Certificate X.509* to select the file from the Explorer path that you would like to import.
  3. Click on the button *Import*.
- ⇒ A success message appears.

## 4 PBX certificates

### 4.1 PBX Certificate

The PBX certificate serves to authenticate the recording server with the PBX. To be able to establish a connection, the PBX must trust the certificate.

There are 3 tabs:

- *About*  
Information about the certificate, display, and usage,  
see [chapter "Tab About", p. 11](#).
- *Current Certificate*  
Information such as version, issuer, validity, IP addresses, DNS, algorithms,  
see [chapter "Tab Current Certificate", p. 11](#)
- *Import Certificate*  
Output keystore path, format selection, path of the certificate,  
see [chapter "Tab Import Certificate", p. 12](#).

#### 4.1.1 Tab About

The tab explains the purpose of the certificate.

The certificate serves to authenticate the recording server whenever the PBX connects with the recording server. The PBX must confirm the certificate. The PBX must trust the certificate to establish a secure connection.

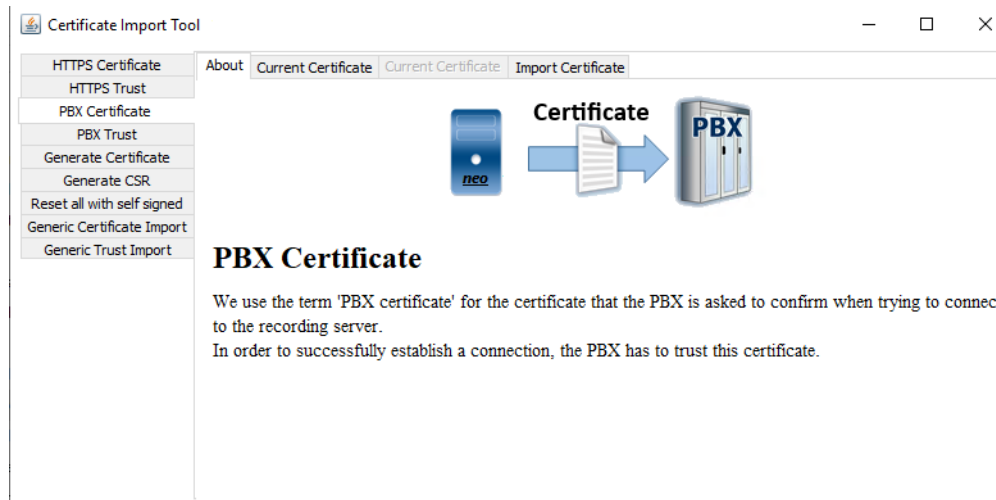


Fig. 8: PBX Certificate - tab About

#### 4.1.2 Tab Current Certificate

This tab displays information about the current certificate of the recording server. You can export the current certificate here.

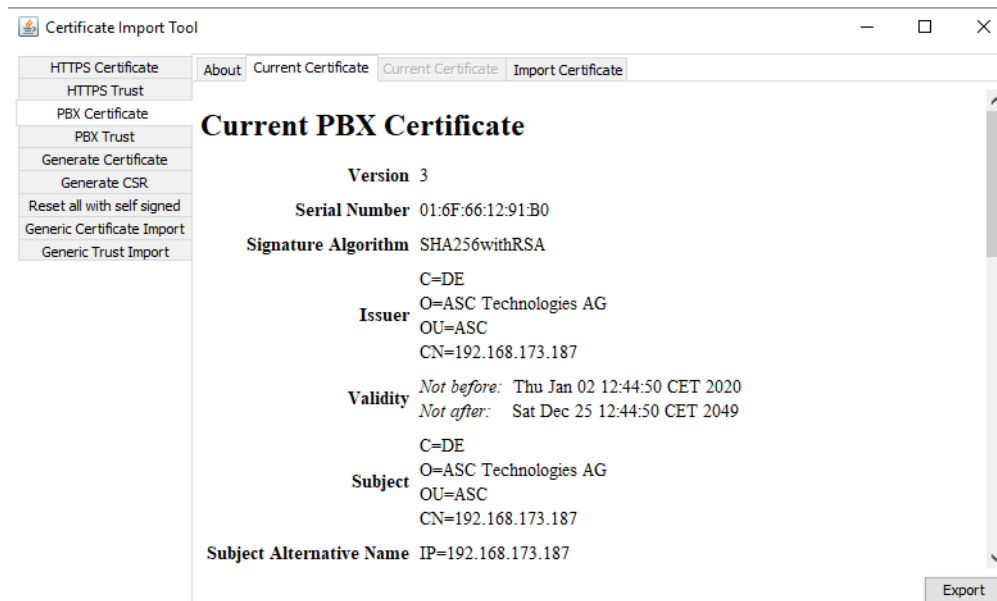


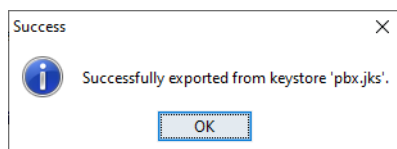
Fig. 9: PBX Certificate - tab Current Certificate

The following information is available:

- *Version*
- *Serial Number*
- *Signature Algorithm*
- *Issuer*
- *Validity*
- *Server details*
- *Alternative server name or DNS*
- *Public Key Algorithm*
- *Public Key Exponent*
- *Public Key*

By clicking on the button *Export*, you can export the certificate.

1. Click on the button *Export* to export the current self-signed certificate.
2. Select an appropriate storage location for the certificate.
3. Click on the button *Save*.
  - ⇒ A success message appears.



#### 4.1.3 Tab Import Certificate

In this tab, you can import the certificate from the PBX.

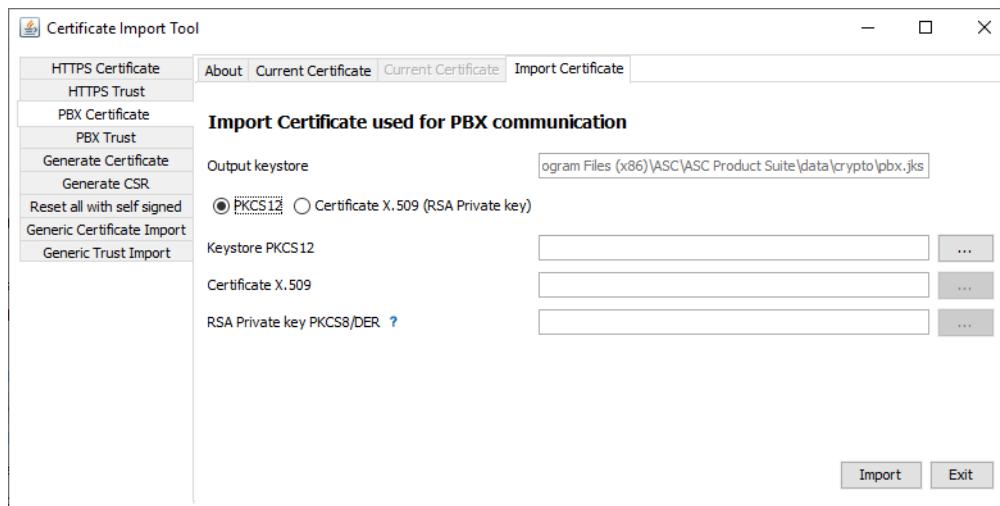



Fig. 10: PBX Certificate - tab Import Certificate

The following formats are supported:

- **PKCS12**  
is a keystore format that contains the public key and the private key in a file.
  - **X.509**  
is the file format of a certificate that contains the public key. If you have requested the X.509 **RSA** certificate via a **CSR**, you must not complete the field **RSA Private Key** as the private key already exists on the system.  
If you do not have requested the X.509 **RSA** certificate via a **CSR**, you must import the private key separately by means of the field **RSA Private Key**.
1. The field *Output keystore* contains the path to the directory where external applications can find the certificates.
  2. Select the format of the certificate by activating the corresponding radio button.
  3. Click on the button  next to the respective format to select the file from the Explorer path.
  4. Click on the button *Import*.
- ⇒ A success message appears.

## 4.2 PBX Trust

The PBX certificate serves to authenticate the PBX with the recording server. To be able to establish a connection, the recording server must trust the certificate of the PBX.

There are 3 tabs:

- **About**  
Information about the certificate, display, and usage, see [chapter "Tab About", p. 13](#).
- **Current Trusted**  
Information such as version, issuer, validity, IP addresses, DNS, algorithms, see [chapter "Tab Current Trusted", p. 14](#)
- **Import Trusted Certificate**  
Output keystore path, format selection, path of the certificate, see [chapter "Tab Import Trusted Certificate", p. 14](#).

### 4.2.1 Tab About

The tab explains the purpose of the certificate.

The PBX Trust certificate serves to authenticate the PBX with the recording server. When establishing a connection from the PBX to the recording server, the recording server must confirm the certificate. The recording server must trust the certificate to be able to establish a connection.

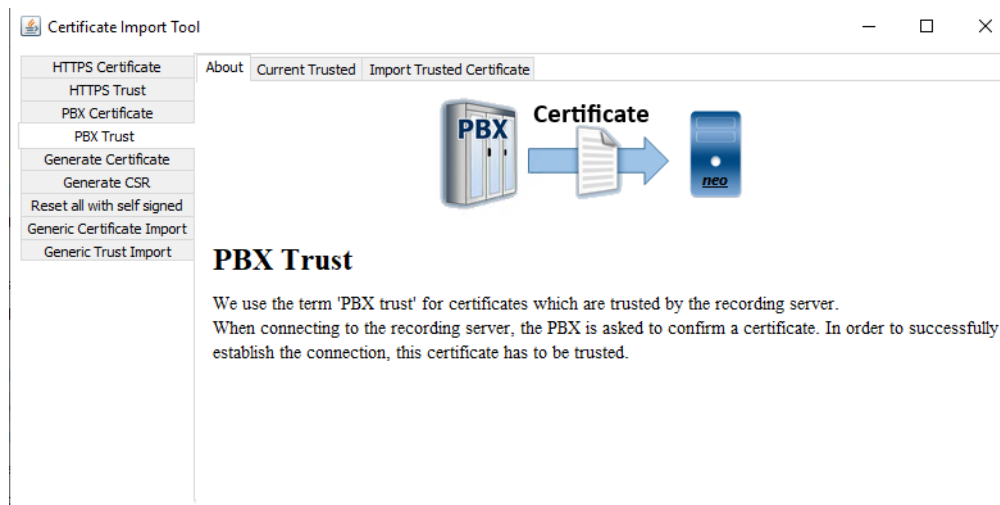


Fig. 11: PBX Trust - tab About

#### 4.2.2 Tab Current Trusted

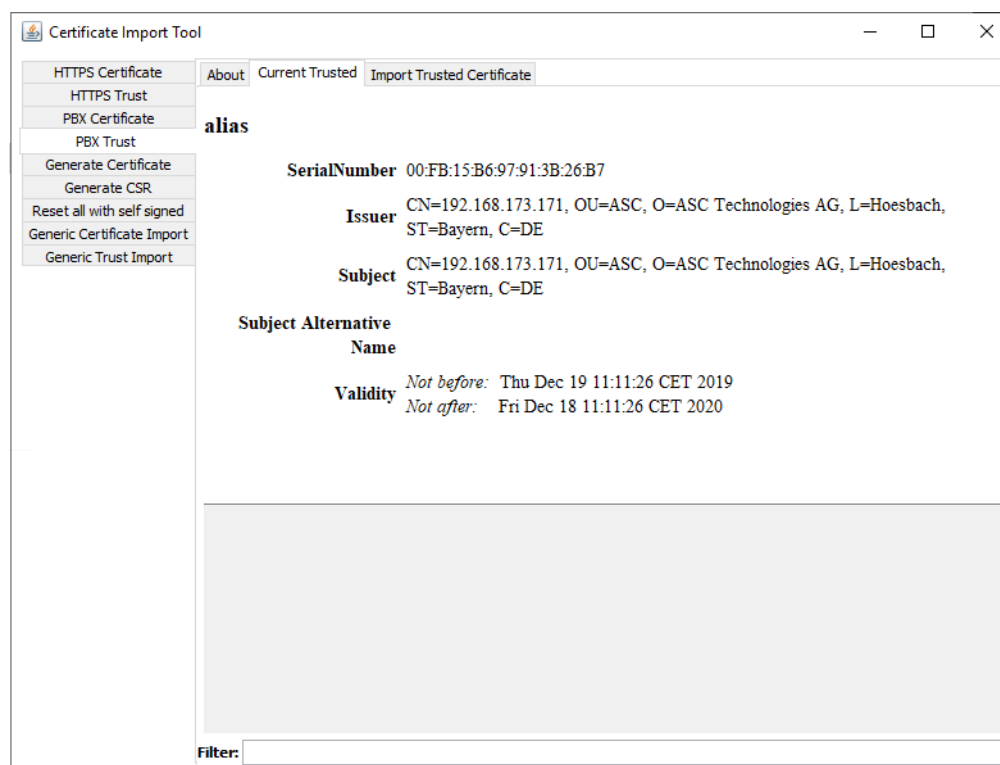


Fig. 12: PBX Trust - tab Current Trusted

#### 4.2.3 Tab Import Trusted Certificate

In this tab, you can import a trusted PBX certificate.

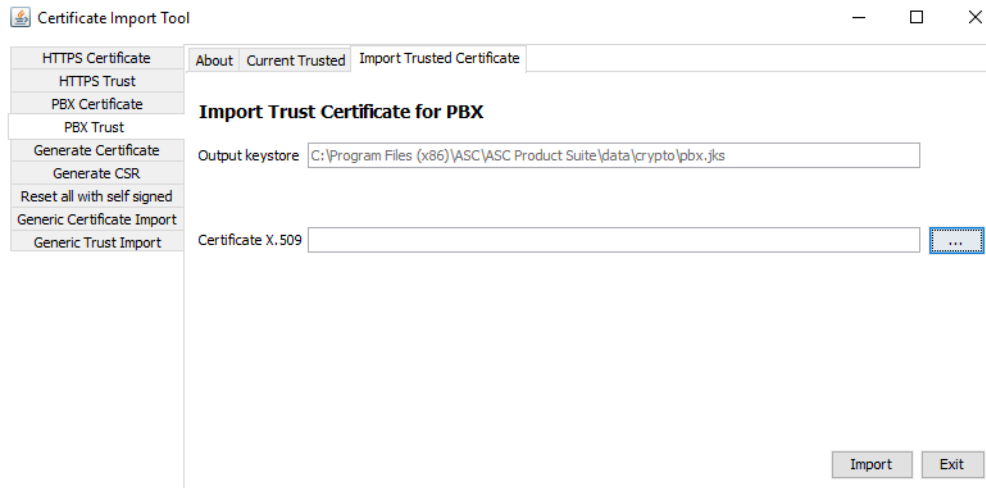


Fig. 13: PBX Trust - tab Import Trusted Certificate

1. The field *Output keystore* contains the path to the directory where external applications can find the certificate.
2. Click on the button ... next to the respective format to select the file from the Explorer path.
3. Click on the button *Import*.  
⇒ The entry dialog for the alias appears.

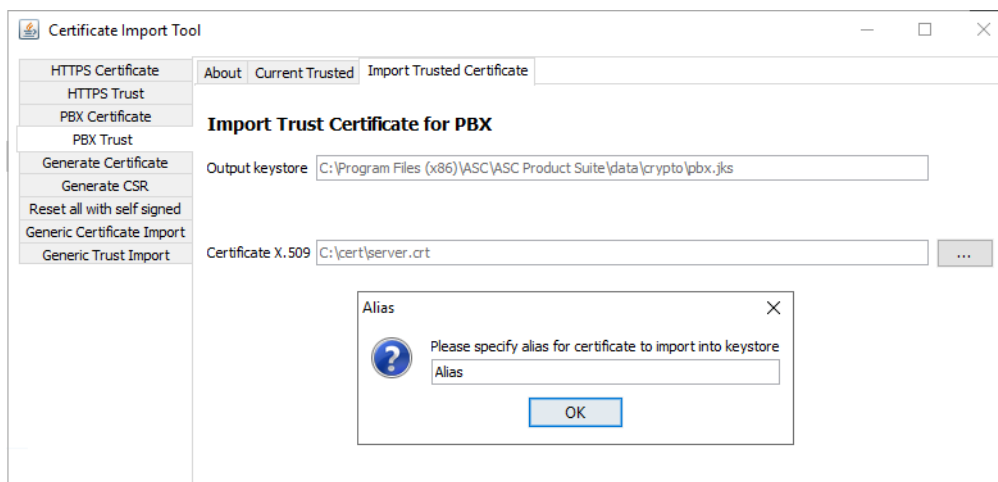


Fig. 14: Entry dialog for alias

4. Enter the alias and click on the button *OK*.  
⇒ A success message appears if the import has been successful.

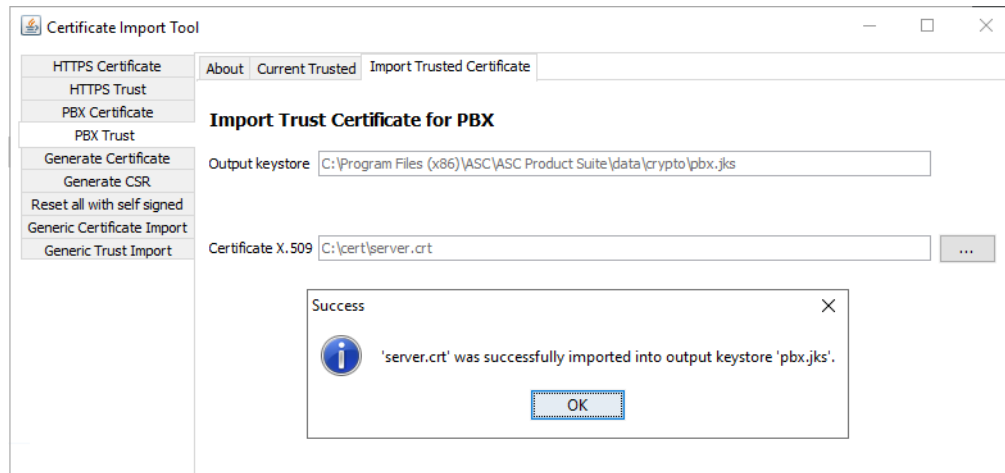


Fig. 15: Success message



## 5

## Generate Certificate

## 5.1

## Generic Certificate Import

In this tab, you can import a certificate to a selected keystore. This must not necessarily be the keystore of the Neo system.

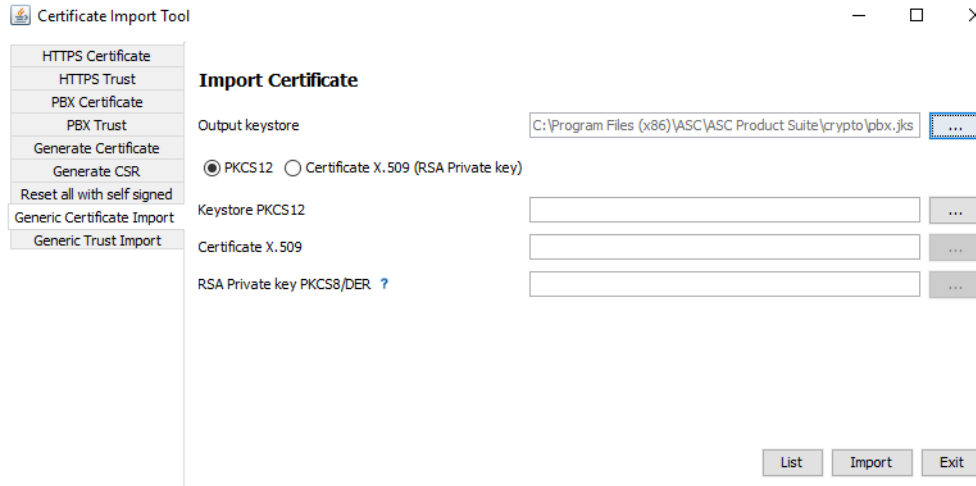


Fig. 16: Generate Certificate - Generic Certificate Import

1. Click on the ... next to the field *Output keystore* to select the directory where external applications can find the certificates.
  2. Select the format of the certificate by activating the corresponding radio button.
  3. Click on the button ... next to the respective format to select the file from the Explorer path.
  4. Click on the button *Import*.
- ⇒ A success message appears.

## 5.2

## Generic Trust Import

In this tab, you can import a [CA](#) certificate to a selected keystore. This must not necessarily be the keystore of the Neo system.

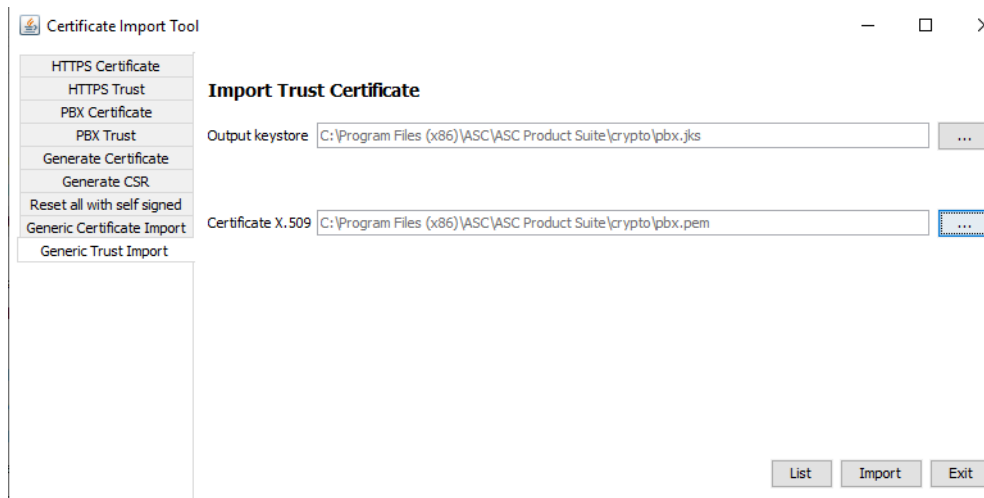


Fig. 17: Generate Certificate - Generic Trust Import

1. Click on the ... next to the field *Output keystore* to select the directory where external applications can find the [CA](#) certificate.
2. Click on the button ... next to the field *Certificate X.509* to select the file from the Explorer path.

3. Click on the button *Import*.
- ⇒ A success message appears.

## 6

## Generate Request

## 6.1

## Generate Certificate

In this tab, you can generate a self-signed certificate. This certificate is not trusted by default. But you can optionally import the file to the trusted keystore so that it is considered trusted in the internal network.

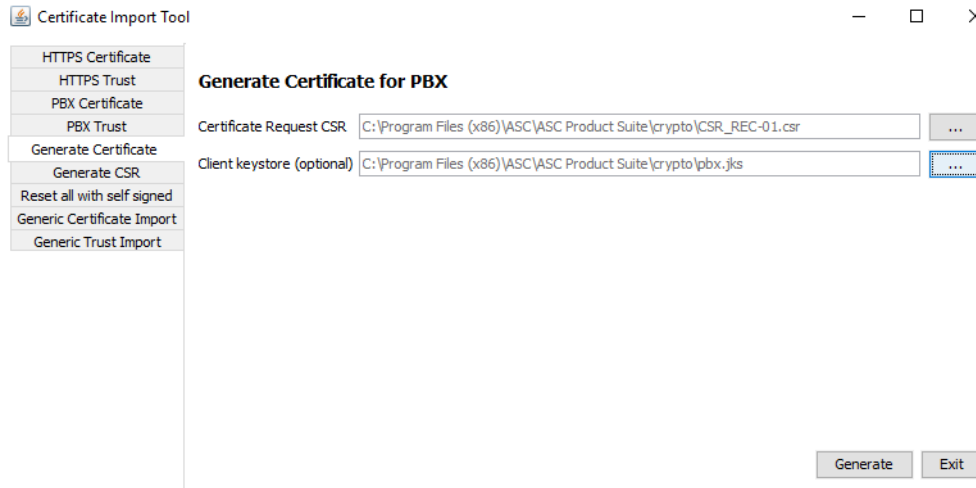


Fig. 18: Generate Certificate

1. Select the menu item *Generate Certificate* in the navigation bar.
2. In the field *Certificate Request CSR*, select the previously created [CRS](#) file by clicking on the button `...`.
3. Select the path to the client keystore in the field *Client keystore*.  
You find the file on the recording server in the directory  
`C:\Program Files (x86)\ASC\ASC Product Suite\data\crypto`.
4. Click on the button *Generate* to confirm the entries and to generate the certificates.
5. Enter the password for the keystore.
6. Click on the button *OK*.  
⇒ The generated certificates and the extended keystore are saved in a ZIP file.
7. Enter a file name for the ZIP file.
8. Click on the button *Save*.  
⇒ The ZIP file is saved.  
⇒ You may unpack the ZIP file in any place you like.

The ZIP file contains 3 files:

File	Content
<i>ca.crt</i>	Public ASC <a href="#">CA</a> certificate.  The CA certificate is a self-signed certificate. But you can import the file to the trusted keystore so that it is considered trusted in the internal network.
<i>cert.crt</i>	Server certificate generated on basis of the certificate request.
<i>pbx.jks</i>	PBX client keystore including the added <a href="#">CA</a> certificate.  .jks format is a proprietary Java format containing information about the imported certificates.

## 6.2

## Generate CSR

In this tab, you can request a certificate from a CA certification authority.

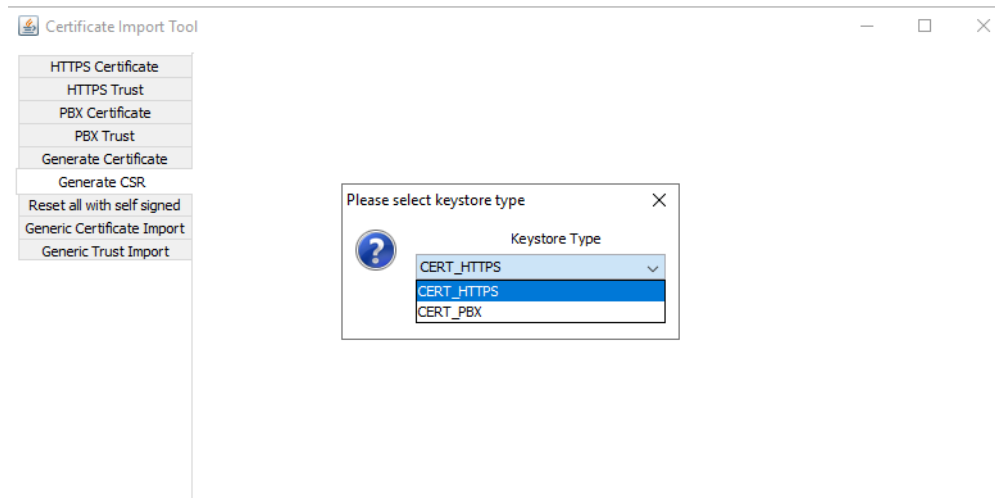


Fig. 19: Generate CSR - request certificate from CA certification authority

1. Select the type of the certificate from the drop-down list.

The following options are available:

- *CERT\_HTTPS*
  - *CERT\_PBX*
2. In the following window, enter the information for the certificate:
    - *Common Name*
    - *Business name*
    - *Department Name*
    - *Town/City*
    - *Province*
    - *Country*
    - *Email address*
    - *Subject alt names (DNS)*
    - *Subject alt names (IP)*
  3. Click on the button *OK* to apply the entries.
  4. In the dialog window, select the storage location and enter a name for the file.
    - ⇒ A success message appears.

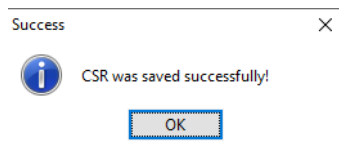


Fig. 20: Generate CSR - save file

You can now send the generated [CSR](#) file to the certification authority.

You can import the certificate that is sent back by the certification authority by selecting the corresponding menu item:

- *CERT\_HTTPS* see [chapter "Tab Import Certificate", p. 7](#)
- *CERT\_PBX* see [chapter "Tab Import Certificate", p. 12](#)

## Reset all with self signed

In this menu item, you can reset all imported certificates to the status of self-signed certificates.



Existing connections which had been considered as trusted by other systems due to the previously configured certificates, no longer work then. As a result, a warning is displayed when opening the requested site in the browser. This also holds true for [PBX](#) connections and other [HTTPS](#) connections.

1. If you would like to overwrite all deployed certificates and private keys, confirm the following security prompt.

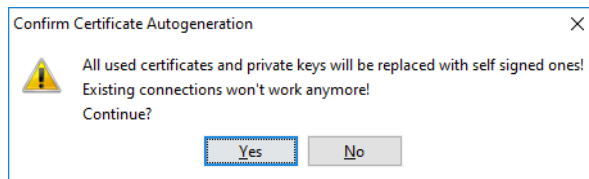


Fig. 21: Security prompt to reset all certificates

## Call up Certimporter in command line

Some functions of the Certificate Import tool may not be executed in the GUI. Instead, you must call up the *certimporter.exe* in the command line.

1. Open PowerShell in the following path:  
C:\Program Files (x86)\ASC\ASC Product Suite\scripts>
2. Execute the following command:  
java -jar .\certimporter.exe -h  
⇒ Help, functions, and available parameters are displayed.

```

-----
DESCRIPTION
-----
--interactive           : Ask alias and password in interactive mode
--autogenerate          : Reset all certificates with self signed
--winTrust              : Import autogenerated certificates as trusted root into windows certificate store
                        : Also rewrites the links on the desktop to refer to IP used in the certificate
--force                 : Force file overwriting (Must be first parameter)

--https                 : Mode for HTTPS certificates
--https-trust           : Mode for HTTPS Trust certificates
--pbx                   : Mode for PBX certificates
--pbx-trust             : Mode for PBX Trust certificates
--transmission          : Mode for Transmission certificate

--list                  : List certificates
--import                : Import certificate
--importalias <string>  : Import given certificate as alias into keystore

--generate              : Generate certificate
--auto                  : Auto generate mode
--cert-csr              : Generate certificate from CSR
--cert-key              : Generate certificate and key
--validity <int>        : Validity in years of auto generated certificate
--ip-address            : IP Address to autogenerate certificate for
--altNames              : A list of alternate names for the machine (SAN) separated by spaces
--export-x509 <path>    : Export path for generated X509 Certificate
--export-pkey <path>    : Export path for generated PKCS8 private key
--csr <path>            : Path to CSR file

--x509 <path>           : Path to X509 certificate
--pkcs12 <path>         : Path to PKCS12 keystore
--pkcs12pw <string>     : PKCS12 keystore password
--pkcs12alias <string>  : Alias for certificate inside PKCS12 keystore
--privatekey <path>    : Path to PKCS8 private key
--privatekeypw <string> : PKCS8 Private key password

Convert your private key to PKCS8 with 'openssl pkcs8 -topk8 -outform DER -in <paht> -out <path>'

Invocations with the --interactive flag will query the user for parameters like aliases, passwords,
ipAddresses and the validity in years.
-----

```

Fig. 22: Description of Certimporter functions

```

-----
USAGE
-----

Autogenerate mode
  Shorthand for "--generate --auto". See below for more details.

Generate mode
  Generate a selfsigned certificate for the neo server and import it for all certificate types.
  "--generate --auto [--winTrust] --ip-address <ipAddress> [--validity <years>] [--altNames <name>
  [2ndName] ...]"
  "--generate --auto [--winTrust] [--altNames <name> [2ndName] ...] --interactive"

  Generate a signed certificate from the specified certification request, using the current pbx key and
  certificate as certification authority.
  "--generate --cert_csr --csr <filePath> {--https|--pbx-trust|--pbx|--pbx-trust|--transmission}"

  Generate a selfsigned certificate and private key.
  "[--force] --generate --cert_key --export-x509 <outCertPath> --export-pkey <outKeyPath> --ip-address
  <ipAddress> [--validity <years>] [--altNames <name> [name ...]]"

List mode
  List current certificate details for the specified type.
  "--list {--https|--pbx-trust|--transmission|--pbx|--pbx-trust}"

Import mode
  Import a certificate to add to the trust store of the specified type under the given alias.
  "--import {--https-trust|--pbx-trust} --x509 <filePath> --importalias <alias>"
  "--import {--https-trust|--pbx-trust} --x509 <filePath> --interactive"

  Import a certificate and associated private key for the given type from separate files.
  "--import {--https|--pbx|--transmission} --x509 <filePath> --privatekey <filePath> [--privatekeypw
  <password>]"
  "--import {--https|--pbx|--transmission} --x509 <filePath> --privatekey <filePath> --interactive"

  Import a certificate and associated private key for the given type from one pkcs12 keystore.
  "--import {--https|--pbx|--transmission} --pkcs12 <filePath> --pkcs12alias <alias> --pkcs12pw <password>"
  "--import {--https|--pbx|--transmission} --pkcs12 <filePath> --interactive"

PS C:\Program Files (x86)\ASC\ASC Product Suite\scripts>

```

Fig. 23: Using Certimporter tool by means of command line

## List of figures

Fig. 1	Start Certificate Import Tool .....	5
Fig. 2	HTTPS Certificate - tab About .....	6
Fig. 3	Information about the current certificate .....	7
Fig. 4	HTTPS Certificate - Tab Import Certificate .....	8
Fig. 5	HTTPS Trust - tab About .....	9
Fig. 6	HTTPS Trust Certificate - tab Current Trusted .....	9
Fig. 7	HTTPS Trust - tab Import Trusted Certificate .....	10
Fig. 8	PBX Certificate - tab About .....	11
Fig. 9	PBX Certificate - tab Current Certificate .....	12
Fig. 10	PBX Certificate - tab Import Certificate .....	13
Fig. 11	PBX Trust - tab About .....	14
Fig. 12	PBX Trust - tab Current Trusted .....	14
Fig. 13	PBX Trust - tab Import Trusted Certificate .....	15
Fig. 14	Entry dialog for alias .....	15
Fig. 15	Success message .....	16
Fig. 16	Generate Certificate - Generic Certificate Import .....	17
Fig. 17	Generate Certificate - Generic Trust Import .....	17
Fig. 18	Generate Certificate .....	19
Fig. 19	Generate CSR - request certificate from CA certification authority .....	20
Fig. 20	Generate CSR - save file .....	20
Fig. 21	Security prompt to reset all certificates .....	21
Fig. 22	Description of Certimporter functions .....	22
Fig. 23	Using Certimporter tool by means of command line .....	23





---

List of tables

## Glossary

### CA

Certification Authority. CA issues X.509 certificates for different purposes.

### CRS

Certificate Signing Request, request by an authorized certification authority to generate a certificate.

### CSR

Certificate Signing Request

### HTTPS

Hypertext Transfer Protocol Secure (HTTPS) is an extension of the Hypertext Transfer Protocol (HTTP). It is used for secure communication over a computer network, and is widely used on the Internet. In HTTPS, the communication protocol is encrypted using Transport Layer Security (TLS), or, formerly, its predecessor, Secure Sockets Layer (SSL). (Source: Wikipedia 23rd October 2019)

### LDAP

Lightweight Directory Access Protocol

### PBX

Private Branch Exchange

### RSA

RSA is one of the first public-key cryptosystems and is widely used for secure data transmission. In such a cryptosystem, the encryption key is public and it is different from the decryption key which is kept secret (private). In RSA, this asymmetry is based on the practical difficulty of the factorization of the product of two large prime numbers, the "factoring problem". [1] A user of RSA creates and then publishes a public key based on two large prime numbers, along with an auxiliary value. The prime numbers must be kept secret. Anyone can use the public key to encrypt a message, but with currently published methods, and if the public key is large enough, only someone with knowledge of the prime numbers can decode the message feasibly. (Source: Wikipedia 24th April 2018)

### Truststore

Collection of certificates that are considered as trusted.