SSL/TLS/HTTPS

Secure communication between components

Version 5.12

Manual

Document: Manual- SSL/TLS/HTTPS Secure communication between components

Release date: 04.05.2023

Document version: 1

Author: FORCAM GmbH

FB_8-41 CREATED ON: 03/20/23 CHANGED ON: 05/04/2023 VERSION: V1
APPROVED BY: HEAD OF TTE AUTHOR: TECHNICAL WRITER CLASSIFICATION: PUBLIC

Contents

1 About this manual

This manual describes the requirements and configuration necessary to implement SSL encryption for the communication within and with FORCE MES FLEX (previously FORCE IIoT, hereafter simply referred to as MES FLEX). It also explains how to provide or create the required certificates.

⚠ This manual is based on FORCE MES FLEX version 5.12.14. Some of the functions mentioned may not be available in earlier FORCE IIOT versions.

ⓘ For better readability, we generally use the generic masculine in the text. These formulations, however, are inclusive of all genders and intended to equally address all readers of the document.

1.1 Target group and previous knowledge

In this manual, we assume that you have knowledge in the use of FORCAM FORCE IIOT.-If you do not have any knowledge in this area, take the time to familiarize yourself with the basics.

ⓘ We recommend that you use our Academy: s**https://forcam.com/academie/**
The FORCAM Academy provides the knowledge to effectively use the methods for digital transformation and the technologies for the Smart Factory.
Based on lean manufacturing and TPM methods, our institute team will guide you to initiate changes in the company and to use the technologies correctly.

2 Secure network communication

In order to make the individual functions and components of MES FLEX available to users, the components must communicate with each other as well as with external applications located within the company network. If there aren't proper security measures in place, any communication within a computer network can be intercepted, read, or modified.
The use of an SSL protocol is a recommended way to secure communication within a network.

ⓘ TLS stands for Transport Layer Security and is the successor protocol to SSL. Both are encryption protocols for the transport layer of the Internet.

2.1 Communication via SSL/TLS

SSL (Secure Socket Layer) as well as TLS (Transport Layer Security) are both industry-standard security technologies that are used to establish an encrypted connection between a web server and a web client (such as a web browser). They are encryption protocols for the transport layer of the Internet. They ensure secure network communications by authenticating the server, thereby ensuring privacy and integrity of all transmitted data.

HTTP
SSL
HTTPS
Certificate

Fig. 1: Overview – Encryption via SSL

If communication takes place via this encrypted transport layer, an "s" is added to the protocol name: Http becomes https, imap becomes imaps, etc.
If a website wants to communicate via https (as the vast majority of websites do nowadays), a so-called SSL certificate is required.

2.2 Certificates

To create an encrypted connection between a web server and a client, an SSL/TLS certificate for the web server is required. This is a digital file with information about the identity of the web server. It also contains the cryptographic key material that is used for securing the communication channel between server and client. A certificate must be created and digitally signed by the owner of the website (domain).
Depending on their trustworthiness, three different types of certificates can be used, which are described in more detail in the following chapters:

- **CA-signed certificates**
  These have the highest trustworthiness, as they have been verified by an independent body.
- **Self-signed certificates within the own domain**
  For this type of certificates, a Windows server acts as root certificate authority (CA).
- **Self-signed certificates**
  Users can create this type of certificates themselves. It should, however, only be used for testing purposes.

ⓘ Chapter 2.2.4 contains an overview of the different certificate types and the recommended areas of use with MES FLEX.

2.2.1 CA-signed certificates

ⓘ Generally recommended by FORCAM for productive systems. Absolutely necessary for web communication over the Internet.

An SSL certificate can be signed not only by the website owner but also by an independent **Certificate Authority** (CA). A CA is a trusted third-party provider that confirms the authenticity of a website. If a website is considered a trusted website, the CA adds its own **digital signature** to the self-signed SSL certificate of that website. This way, web clients can be sure that the identity of the website has been verified.
If an SSL certificate issued by a known CA is used, secure communication between the server and the web client takes place automatically. No warning message is displayed in the web browser because the website has been verified by the CA.

Communication with and within production systems should always be secured using certificates signed by a certificate authority, especially when users outside the organization access the MES FLEX server via browser clients. If the server is not protected by a firewall, for example, and is accessed over the Internet, a CA-signed certificate guarantees clients outside the organization that the identity of the website has been verified by an independent body.

2.2.2 Self-signed certificates within a domain

ⓘ Recommended by FORCAM for use within the customer’s own domain.

Within a Windows domain, a role called **Active Directory Certificate Services (AD CS)** can be installed on a Windows server as an Enterprise Root Certification Authority (CA).
This way, a root certificate can be created on the server, which **is valid for the entire domain**.

This root certificate can then also be rolled out to the individual clients of the same domain.
Within the client’s domain (MES FLEX system), any self-signed certificate that was created using the root certificate can therefore be considered trustworthy for the client-server connection.

2.2.3 Self-signed certificates

⚠ **Not recommended by FORCAM: Do not use in productive FORCE IIOT/MES FLEX systems.**
A self-signed certificate is an SSL certificate that was signed only by the owner of the website. Self-signed certificates are occasionally used in websites that are only available to users within the organization's internal network (LAN).

Whenever a web client (for example, a web browser) connects to a website using a self-signed SSL certificate, a warning informs the user that the website could not be verified as a trusted website. Depending on the browser type, there are settings to suppress these warnings about self-signed certificates.

2.2.4 Overview: Certificates and areas of use

Certificate type	Area of use	Central features
CA signed certificate	Productive systems with and without web communication Absolutely necessary whenever the Force clients are connected via the Internet	– The trustworthiness is validated by CA certification (i.e., via an independent trustworthy organization).# – No warning message in the browser, communication automatically secured.
Self-signed certificate in own domain	Production systems where client-server communication takes place only within the own domain.	– Root certificates created by your own Windows server are valid for the entire domain. – Trustworthiness is guaranteed within the internal network (MES-FLEX system).
Self-signed certificates	Test systems	– Secure network communication (LAN) ensured only within the organization (not via the Internet)

2.3 Keystore

A keystore is a repository of certificates. The keystore contains the certificates that ensure secure data transmission via TLS and Public Key Infrastructure (PKI). The public certificate of the service and the corresponding private key are also stored there. When MES FLEX is installed or updated, FFSetup takes over certificate administration in the keystore (see chapter 4.1 and 4.2).

3 Requirements on the customer side

3.1 Provision of certificates

ⓘ **Customer responsibilities**:

- Provide certificate(s) for MES FLEX
- Update certificate(s) - valid runtime

The customer's internal IT department must provide a valid certificate for MES FLEX. If no internal know-how is available within the company, an external service provider can be hired to provide the certificates.

The following requirements apply for MES FLEX certificates:

General requirements

Note the following general requirements for certificate creation:

- The SSL certificate was created and digitally signed by the owner of the website. The signing certificate authority (CA) must be considered trustworthy by the MES FLEX clients.
- The host names are located in the customer’s domain and can be resolved via the domain name service (DNS).
- An individual certificate is created for each MES FLEX instance.

Technical requirements

The generated certificate must comply with the following technical requirements:

Table 2: Technical requirements for the MES FLEX certificate

Hostname (DNS)	The certificate contains the host name used (i.e., the server on which MES Flex is installed).
Subject Alternative Name (SAN)	Must contain the "FQDN" (Full Qualified Domain Name).
Certificate validity	Each certificate has an expiration date in the future (max. 1 year).
Intended use	The certificate is designed to be used with a web server.
Minimum key length	For RSA = 2048 bit For ECDSA = 256 bit
Hash algorithm	SHA-256 or newer/better
File format	PKCS #12" file format and "pfx" file extension The certificate files should be available in PKCS #12 file format. To avoid compatibility issues, a current Java 11 version must be used according to system requirements.
Password	The password for the certificate files must be provided to FORCAM.

4 Configuration

4.1 New installation (FFSetup)

The installation with FFSetup creates the SSL/HTTPS configuration for the application.

4.1.1 Port configuration

During installation, default ports are entered in the properties file (common.properties) for the following components and services:

Standard components:

- Shopfloor Terminal (SFT)
- Workbench
- NewOffice

Additional Services:

- ffauth
- fferp
- ffwebservices

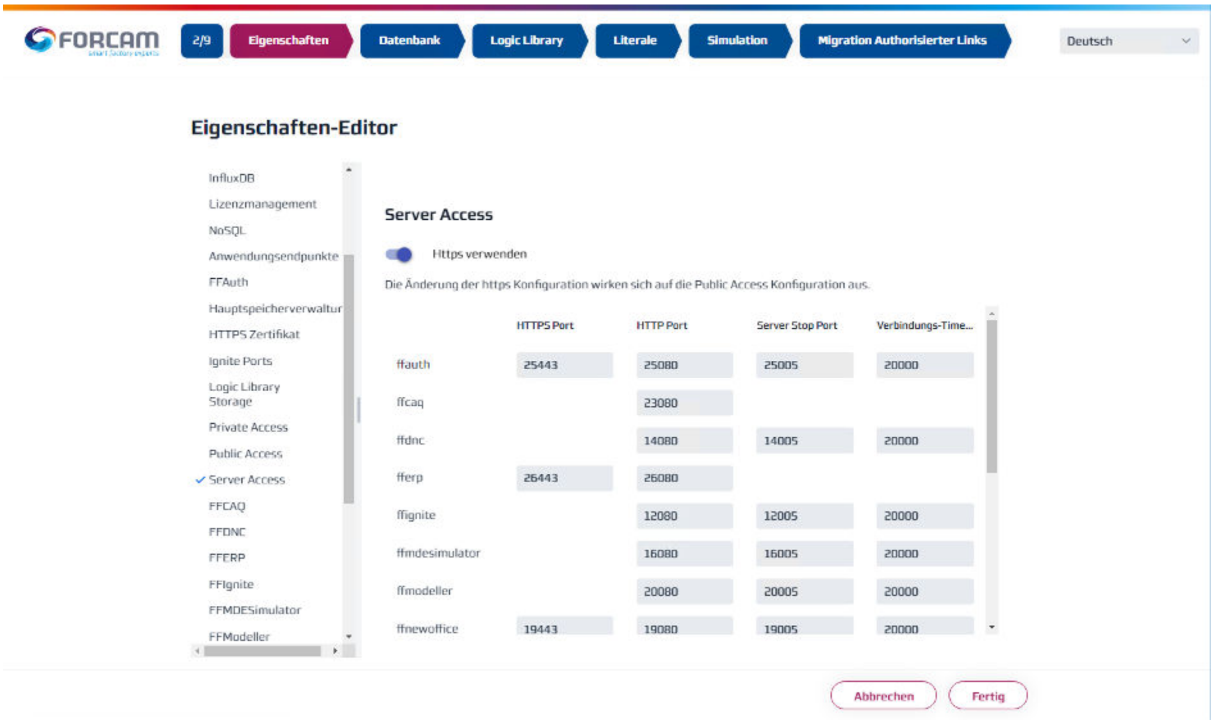
ⓣ **Example**:
For the Workbench, a private port 15080 is defined for the HTTP protocol and a public port 15443 for the HTTPS protocol.

The specified ports are the default configuration for MES FLEX.
Changes to the default settings are saved in the "customized.properties" file (see chapter 4.3). This makes the settings available for FFSetup in the next update.

⚠ If you change the default configuration, make sure the specified ports have the same properties (see the screenshots of the individual access types below).

Server access ports

The following figure shows the ports of the services that are accessible and open.



Server Access

● Https verwenden

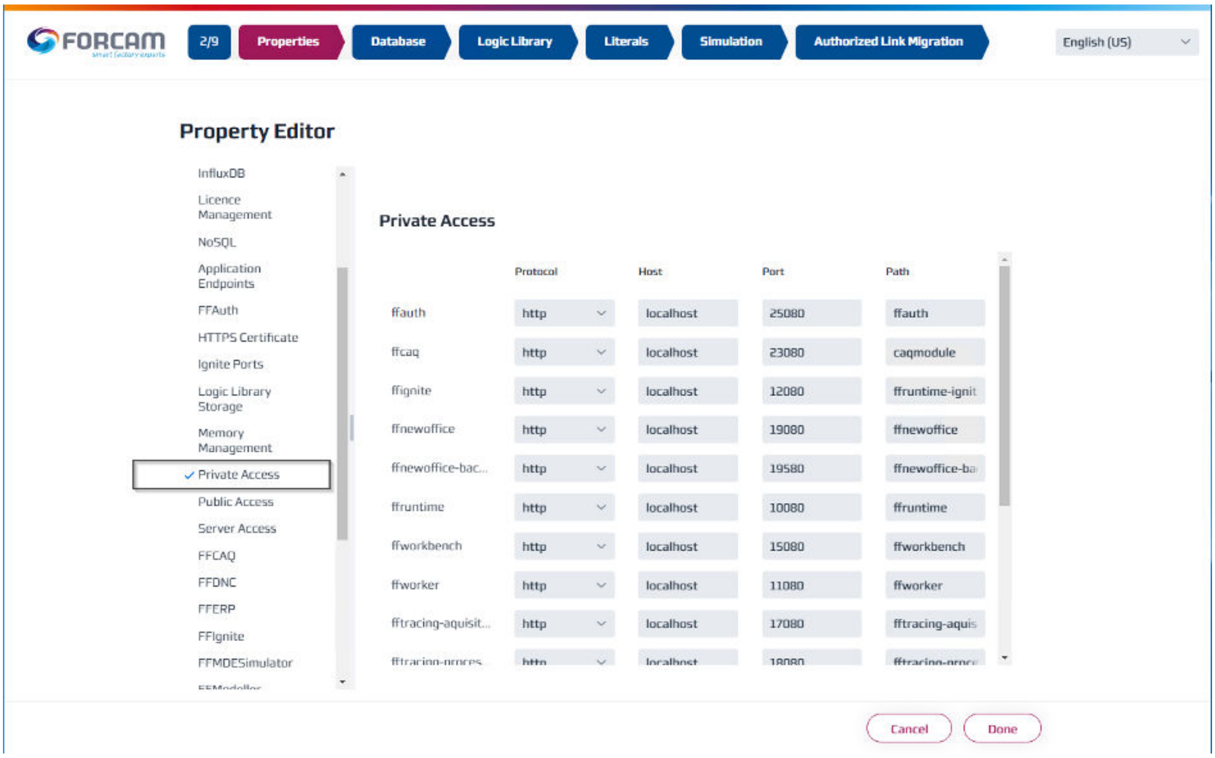
Die Änderung der https Konfiguration wirken sich auf die Public Access Konfiguration aus.

	HTTPS Port	HTTP Port	Server Stop Port	Verbindungs-Time...
ffauth	25443	25080	25005	20000
ffcaq		23080		
ffdnc		14080	14005	20000
fferp	26443	26080		
ffignite		12080	12005	20000
ffmdesimulator		16080	16005	20000
ffmodeller		20080	20005	20000
ffnewoffice	19443	19080	19005	20000

Fig. 2: Server access (default configuration for Tomcat)

Private access ports

These ports are used for communication between the services.



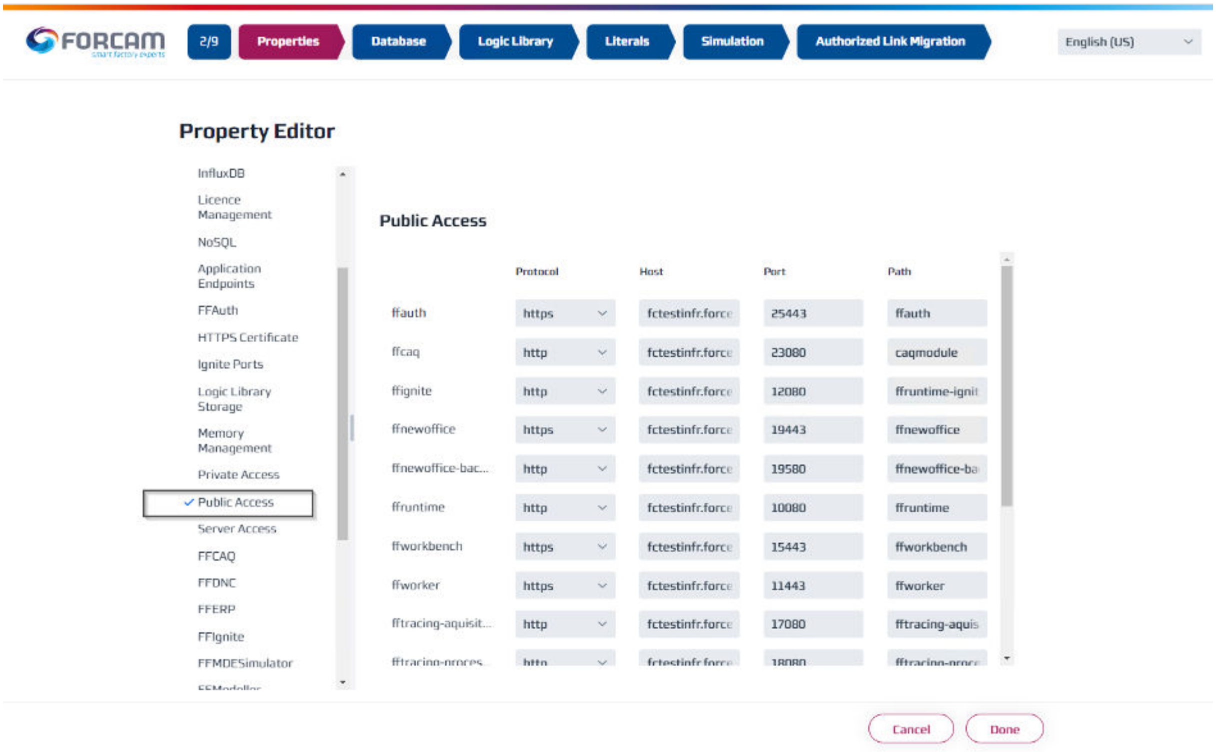
Private Access

	Protocol	Host	Port	Path
ffauth	http	localhost	25080	ffauth
ffcaq	http	localhost	23080	caqmodule
ffignite	http	localhost	12080	ffruntime-ignit
ffnewoffice	http	localhost	19080	ffnewoffice
ffnewoffice-bac...	http	localhost	19580	ffnewoffice-ba
ffruntime	http	localhost	10080	ffruntime
ffworkbench	http	localhost	15080	ffworkbench
ffworker	http	localhost	11080	ffworker
fftracing-aquisit...	http	localhost	17080	fftracing-aquis
fftracing-proces...	http	localhost	18080	fftracing-proce

Fig. 3: Private access ports (default configuration)

Public access ports

These ports are used for client-server communication (Workbench, New Office, SFT).



Property Editor

Public Access

	Protocol	Host	Port	Path
ffauth	https	fctestinfr.force	25443	ffauth
ffcaq	http	fctestinfr.force	23080	caqmodule
ffignite	http	fctestinfr.force	12080	ffruntime-ignit
ffnewoffice	https	fctestinfr.force	19443	ffnewoffice
ffnewoffice-bac...	http	fctestinfr.force	19580	ffnewoffice-ba
ffruntime	http	fctestinfr.force	10080	ffruntime
ffworkbench	https	fctestinfr.force	15443	ffworkbench
ffworker	https	fctestinfr.force	11443	ffworker
fftracing-aquisit...	http	fctestinfr.force	17080	fftracing-aquis
fftracing-proces...	http	fctestinfr.force	18080	fftracing-proce

Fig. 4: Public access ports (default configuration)

4.1.2 HTTPS certificate

After starting FFSetup, the **HTTPS Certificate** page can be accessed in the Property Editor. During a new installation, either an existing certificate can be uploaded, or a new (self-signed) certificate can be generated.

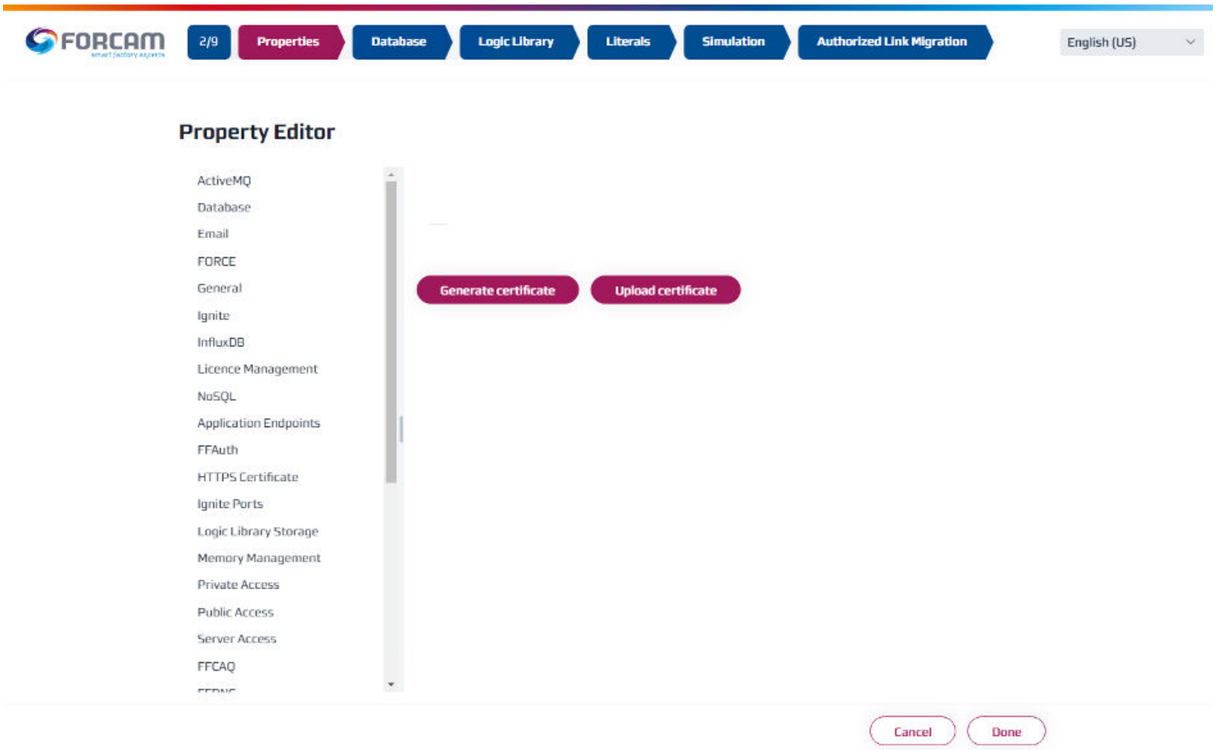
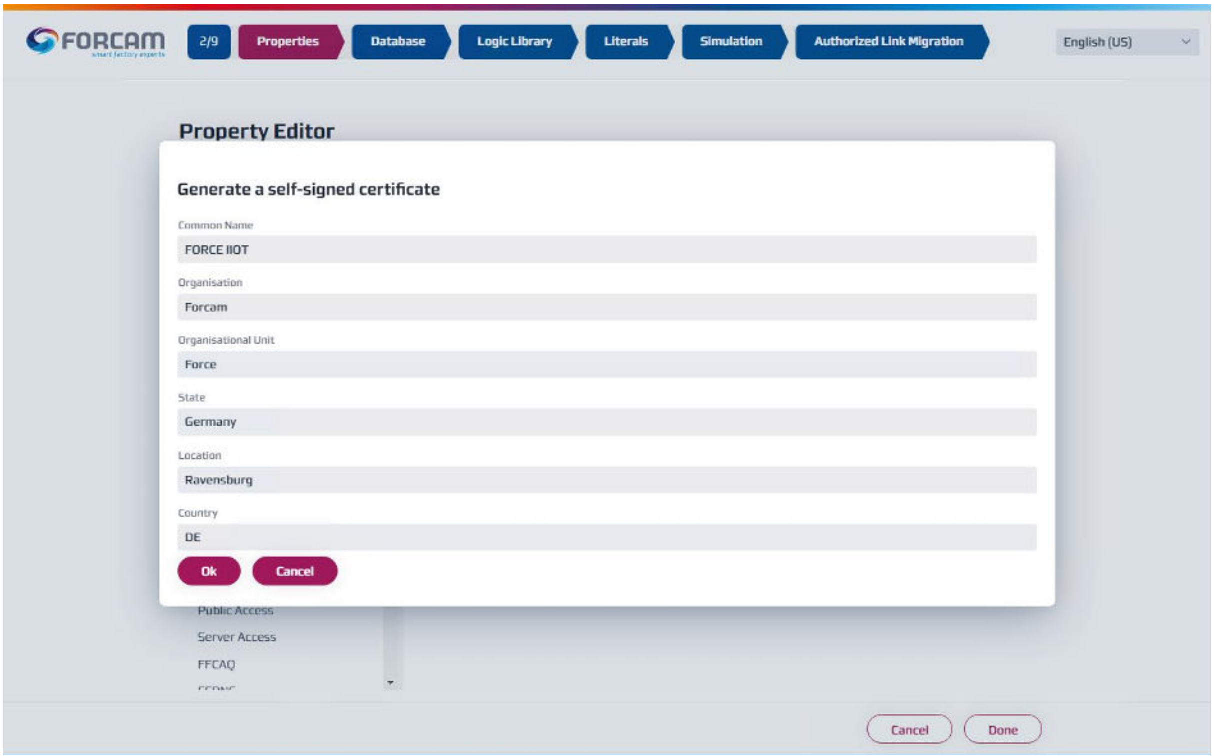


Fig. 5: FFSetup: Generate or upload certificate

Generate certificate

⚠ Do not use with productive MES FLEX systems. The origin of self-signed certificates is not verified by an independent certification body (see chapter 2.2.3 Self-signed certificates).



Generate a self-signed certificate

Common Name: FORCE IIOT

Organisation: Forcam

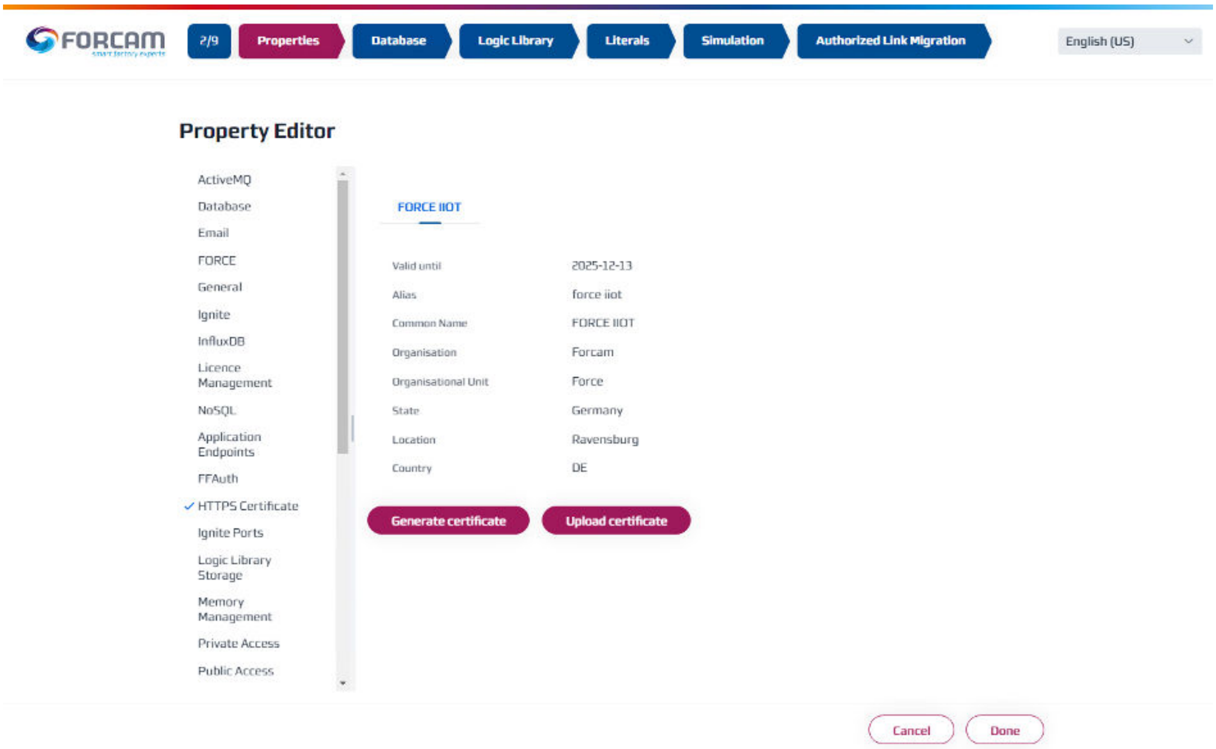
Organisational Unit: Force

State: Germany

Location: Ravensburg

Country: DE

Fig. 6: Generate self-signed certificate (example)

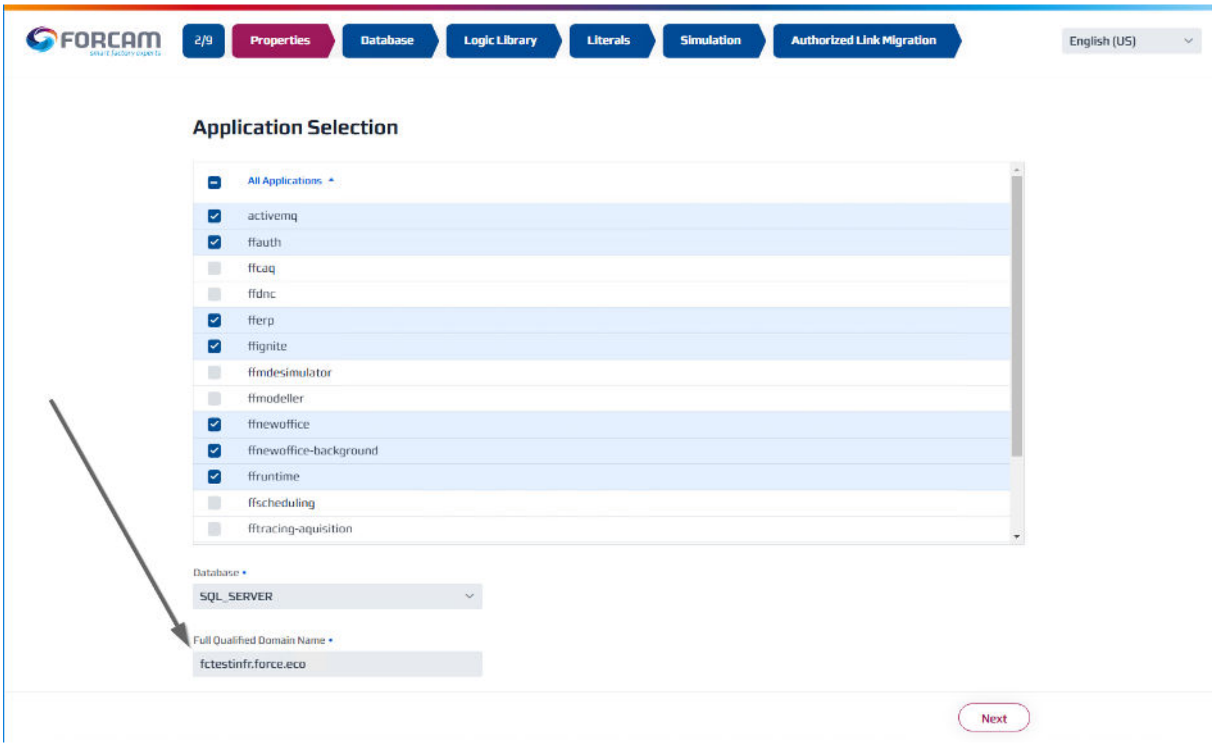


Valid until	2025-12-13
Alias	force iiot
Common Name	FORCE IIOT
Organisation	Forcam
Organisational Unit	Force
State	Germany
Location	Ravensburg
Country	DE

Fig. 7: Generated certificate (example)

Upload certificate

⚠ The host name used (Full Qualified Domain Name) must be identical to the Common Name (CN) on the certificate. See chapter 3.1 for further certificate requirements.



Application Selection

- ☐ All Applications
- ☑ activemq
- ☑ ffauth
- ☐ ffcaq
- ☐ ffdnc
- ☑ fferp
- ☑ ffignite
- ☐ ffmdesimulator
- ☐ ffmodeller
- ☑ ffnewoffice
- ☑ ffnewoffice-background
- ☑ ffruntime
- ☐ ffscheduling
- ☐ fftracing-aquisition

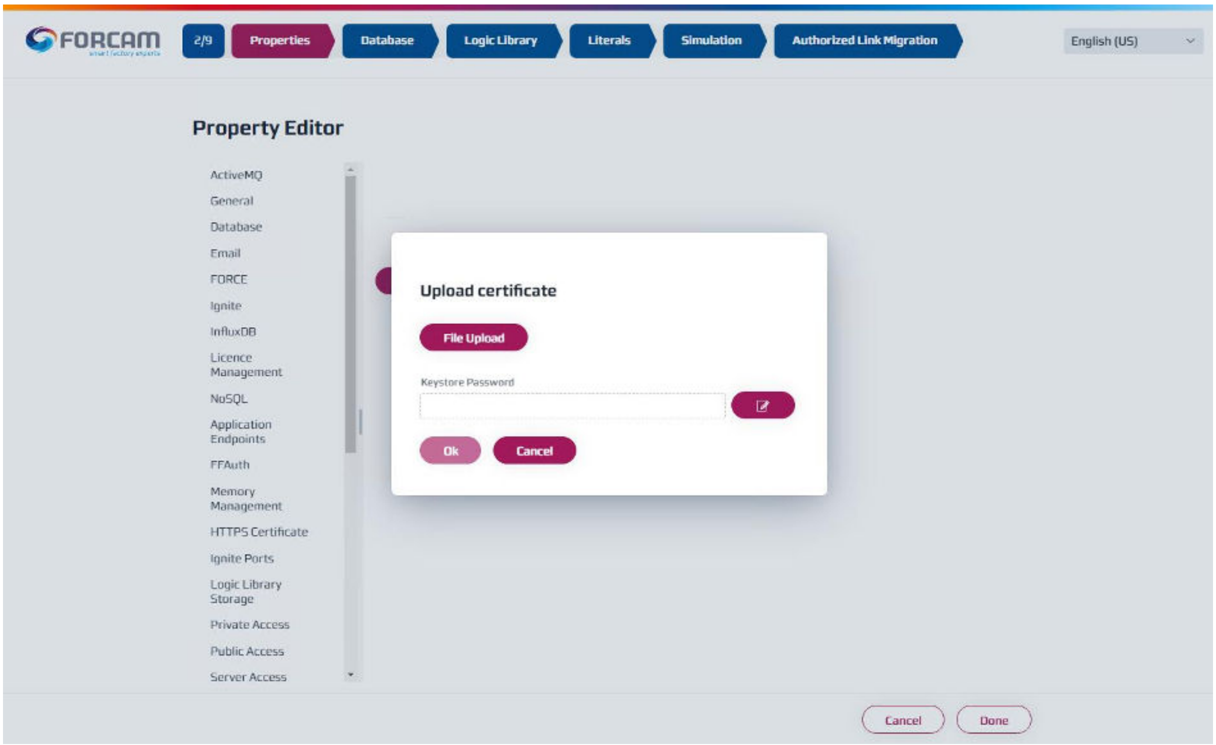
Database *
SQL_SERVER

Full Qualified Domain Name *
fctestinfr.force.eco

Fig. 8: Full Qualified Domain Name

The valid certificate can be uploaded in the dialog.

⚠ Observe the additional requirements for MES FLEX certificates (see chap. 3.1).



Upload certificate

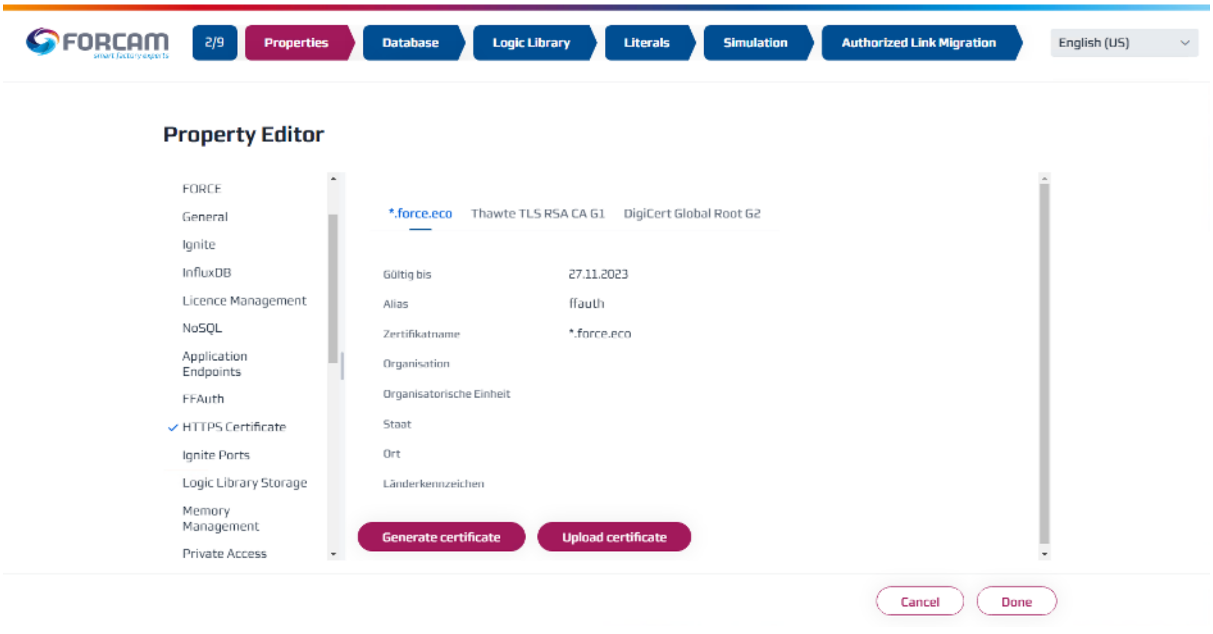
File Upload

Keystore Password

Ok Cancel

Fig. 9: Upload certificate

The uploaded certificate is displayed in the Property Editor:

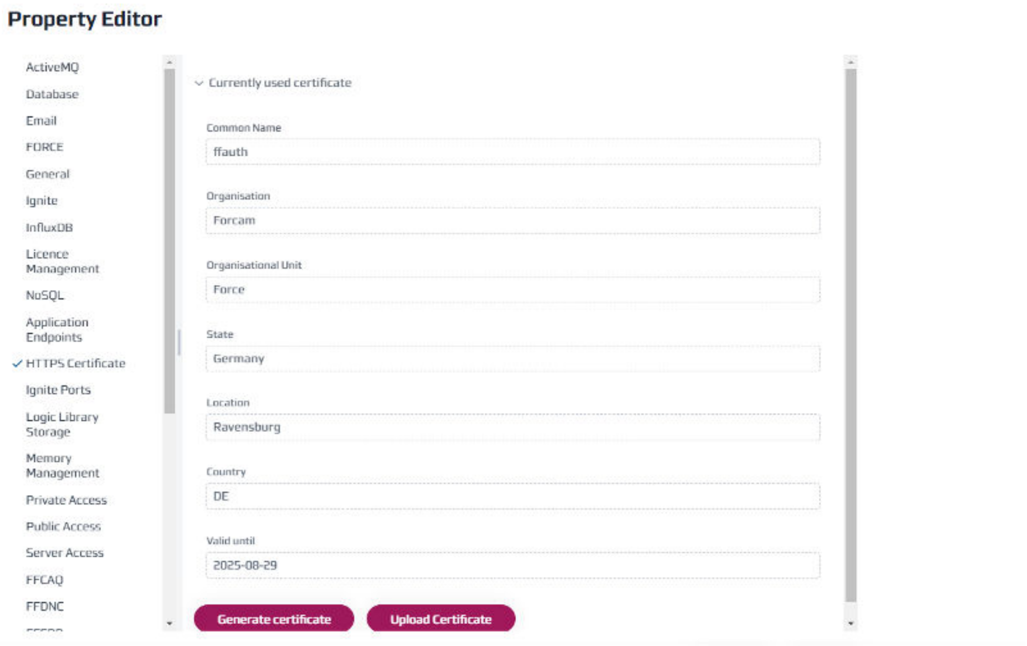


Gültig bis	27.11.2023
Alias	ffauth
Zertifikatname	*.force.eco
Organisation	
Organisatorische Einheit	
Staat	
Ort	
Länderkennzeichen	

Fig. 10: FFSetup: Available certificates

4.2 Update with FFSetup

During an update with FFSetup, the existing certificate is retained:



Currently used certificate

Common Name: ffauth

Organisation: Forcam

Organisational Unit: Force

State: Germany

Location: Ravensburg

Country: DE

Valid until: 2025-08-29

Fig. 11: FFSetup Update: Currently used certificate

ⓘ Changes to the default settings are saved in the "customized.properties" file (see chap. 4.3). This makes the settings available for FFSetup in the next update.

4.3 Directories and files

The following directories and files are relevant for the SSL configuration:

Contents/File	Path and hints
Certificate	D:\ForcamForce\app\config\.keystore
Properties	D:\ForcamForce\appconfig
Common Properties	D:\ForcamForce\app\config\common.properties Example (excerpt): … *#https* *https.enabled=true* … *#keystore* *keystore.password=FQNLul6Qk68h+htMlJbOXlLh1RYnA=* *keystore.path=E:\ForcamForce\app\config\.keystore* … *#ssl* *ssl.certificate=MIILbACCBagwggWkAgEAMIIFnQY……* … *#ffworkbench* *ffworkbench.private.host=localhost* *ffworkbench.private.path=ffworkbench* *ffworkbench.private.port=15080* *ffworkbench.private.protocol=http* *ffworkbench.public.host=SERVERNAME* *ffworkbench.public.path=ffworkbench* *ffworkbench.public.port=15443* *ffworkbench.public.protocol=https* …
Customized Properties	D:\ForcamForce\app\config\customized.properties Changes to the default settings are saved here. This makes the settings available for FFSetup in the next update. Example (excerpt): *#common* … *common.keystore.password=9p424LmjSrwoOdERiVlHgYrvUuM=* *common.keystore.path=F:\ForcamForce\app\config\.keystore*
Tomcat configuration	The "server.xml" are generated during the installation (in the "Application Installation” step)

	D:\ForcamForce\app\workbench-tomcat\server.xml
	Example (excerpt):
	`<!--- HTTPS connector -->`
	`<Connector port="15443"`
	`…`
	`sslProtocol="TLS"`
	`keystoreFile="E:\ForcamForce\app\config\.keystore"`
	`keystorePass="FQNLul6Qk682zx5iRmsMlJbOXlLh1RYnA="`

5 Encryption restrictions

MES FLEX communication endpoints are not encrypted for the following services:

- Message Broker (ActiveMQ)

6 Appendix

6.1 Document conventions

Table 3: Fonts, formatting and characters used

Conventions	Description
Bold type	Buttons and options names are written in bold type.
Italics	Highlighted words are in italics.
Path	Each speficied **Pfad** refers to FORCE MES LITE. The respective module is listed in parentheses.
Values/Quantities	Values/Quantities that are not specified in more detail (e.g. by additions such as target/actual) refer to recorded data.
Icons	For a function that is represented by an icon, the icon is referenced as the object.
Alternative action step	Alternative action steps are separated by Or.
Substeps of an action	Substeps of an action are indented and have unified symbols per action level. The sequence order of the level is: 1. a. i. 1: Etc.
Action result	Action results are indicated by →.
Prerequisites	Prerequisites are indicated by ✓.
Warnings	Warnings are indicated by ⚠.
Notes	Notes are indicated by ⓘ.
Tips	Tips are indicated by ⓣ.

6.2 Abbreviations and terms

Table 4: Abbreviations and terms used

Abbreviation/term	Description
SSL	Secure Socket Layer Encryption protocol used for the Internet transport layer The data streams between client and server are encrypted.
TLS	Transport Layer Security Encryption protocol for the Internet transport layer The data streams between client and server are encrypted. TLS is the successor protocol to SSL.
SFT	Shopfloor Terminal
CA	Certificate Authority
Keystore	Certificate repository

6.3 Table of figures