

APPLICATION EXAMPLE

# AC500 MQTT & MOSQUITTO

## FIRST STEPS AND CONFIGURATION



# Contents

<b>1 Disclaimer</b>	<b>3</b>
<b>2 Abbreviations</b>	<b>4</b>
<b>3 Introduction</b>	<b>5</b>
3.1 Scope of the document	5
3.2 Compatibility	5
3.3 System design & components	5
<b>4 Mosquitto setup &amp; configuration</b>	<b>7</b>
4.1 Install Mosquitto	7
4.2 Start Mosquitto without encryption	8
4.3 Start Mosquitto with encryption	10
4.3.1 Create self-signed certificates	10
4.3.2 Adapt mosquito configuration	14
4.3.3 Start Mosquitto broker	16
<b>5 Mqtt.fx setup &amp; configuration</b>	<b>17</b>
5.1 Install Mqtt.fx	17
5.2 Start Mqtt.fx without encryption	17
5.3 Start Mqtt.fx with encryption	19
<b>6 PLC Configuration</b>	<b>21</b>
6.1 Settings to establish cloud connection	21
6.2 Settings for publishing & subscribing messages	22
<b>7 FAQs</b>	<b>23</b>

# 1 Disclaimer

A. For customers domiciled outside Germany /

Für Kunden mit Sitz außerhalb Deutschlands

**„Warranty, Liability:**

The user shall be solely responsible for the use of this products described within this file. ABB shall be under no warranty whatsoever. ABB's liability in connection with application of the products or examples provided or the files included within this products, irrespective of the legal ground, shall be excluded. The exclusion of liability shall not apply in the case of intention or gross negligence. The present declaration shall be governed by and construed in accordance with the laws of Switzerland under exclusion of its conflict of laws rules and of the Vienna Convention on the International Sale of Goods (CISG)."

**„Gewährleistung und Haftung:**

Der Nutzer ist allein für die Verwendung des in diesem Dokument beschriebenen Produkte und beschriebenen Anwendungsbeispiele verantwortlich.

ABB unterliegt keiner Gewährleistung. Die Haftung von ABB im Zusammenhang mit diesem Anwendungsbeispiel oder den in dieser Datei enthaltenen Dateien - gleich aus welchem Rechtsgrund - ist ausgeschlossen. Dieser Ausschluss gilt nicht im Falle von Vorsatz oder grober Fahrlässigkeit. Diese Erklärung unterliegt Schweizer Recht unter Ausschluss der Verweisungsnormen und des UN-Kaufrechts (CISG)."

B. Nur für Kunden mit Sitz in Deutschland

**„Gewährleistung und Haftung:**

Die in diesem Dokument beschriebenen Anwendungsbeispiele oder enthaltenen Dateien beschreiben eine mögliche Anwendung der AC500 bzw. zeigen eine mögliche Einsatzart. Sie stellen nur Beispiele für Programmierungen dar, sind aber keine fertigen Lösungen. Eine Gewähr kann nicht übernommen werden.

Der Nutzer ist für die ordnungsgemäße, insbesondere vollständige und fehlerfreie Programmierung der Steuerungen selbst verantwortlich. Im Falle der teilweisen oder ganzen Übernahme der Programmierbeispiele können gegen ABB keine Ansprüche geltend gemacht werden.

Die Haftung von ABB, gleich aus welchem Rechtsgrund, im Zusammenhang mit den Anwendungsbeispielen oder den in dieser Datei enthaltenen Beschreibung wird ausgeschlossen. Der Haftungsausschluss gilt jedoch nicht in Fällen des Vorsatzes, der groben Fahrlässigkeit, bei Ansprüchen nach dem Produkthaftungsgesetz, im Falle der Verletzung des Lebens, des Körpers oder der Gesundheit oder bei schuldhafter Verletzung einer wesentlichen Vertragspflicht. Im Falle der Verletzung einer wesentlichen Vertragspflicht ist die Haftung jedoch auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht zugleich ein anderer der in Satz 2 dieses Unterabsatzes erwähnten Fälle gegeben ist. Eine Änderung der Beweislast zum Nachteil des Nutzers ist hiermit nicht verbunden.

Es gilt materielles deutsches Recht unter Ausschluss des UN-Kaufrechts."

## 2 Abbreviations

AB	Automation Builder
CMD	Command Prompt
JSON	JavaScript Object Notation
MQTT	Message Queuing Telemetry Transport
MS	Microsoft
PLC	Programmable Logic Controller

## 3 Introduction

### 3.1 Scope of the document

This manual should give a first introduction into setting up a local Mosquitto broker and send data via MQTT protocol from the AC500 to the local Mosquitto installed on a PC. It provides a step-by-step description of the Mosquitto & PLC configuration for a demo application.

#### Capabilities of demo application

The demo application is able to publish data to a Mosquitto broker, subscribe a topic and receive data from the broker.

### 3.2 Compatibility

The application example explained in this document have been used with the below engineering system versions. They should also work with other versions, nevertheless some small adaptations may be necessary, for future versions.

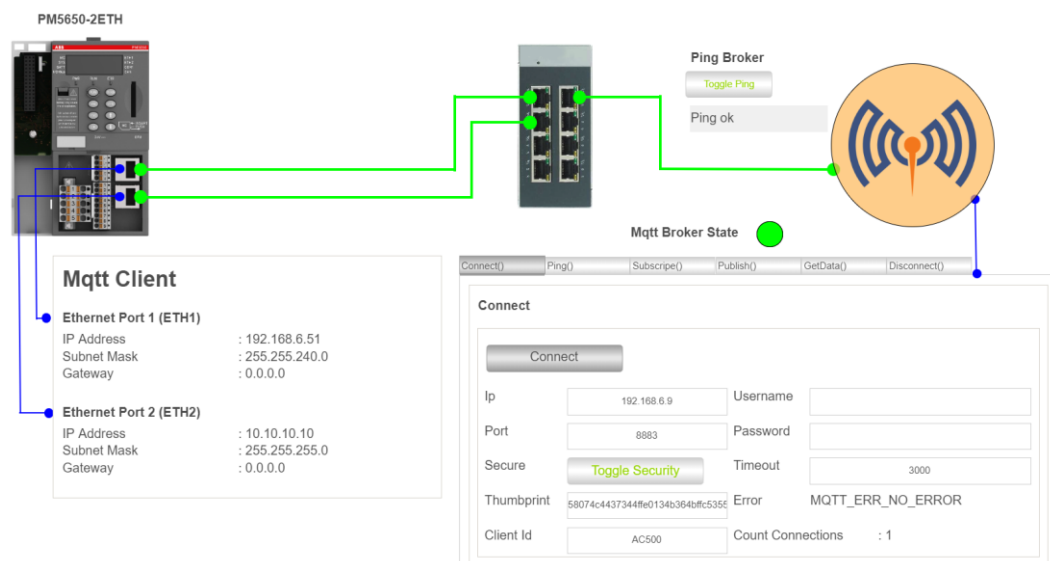
- AC500 PLC (V2 / V3)
- Automation Builder 2.5.0 or newer

### 3.3 System design & components

The system contains the following components:

- AC500
  - PM5650
  - MQTT-Library
- Mosquitto
- Mqtt.fx

The system design is shown in the visualization:



Component	Function
AC500	Acts as MQTT client publishing the data to a local Mosquitto broker using the MQTT library. The data which is converted into JSON-format.
Broker	Here a local Mosquitto broker. Manages connection of Edge-devices, receives messages sent from AC500 and makes them accessible for other MQTT clients.
Mqtt.fx	Mqtt.fx is a graphical MQTT client who can access the broker and publish or subscribe messages.

## 4 Mosquitto setup & configuration

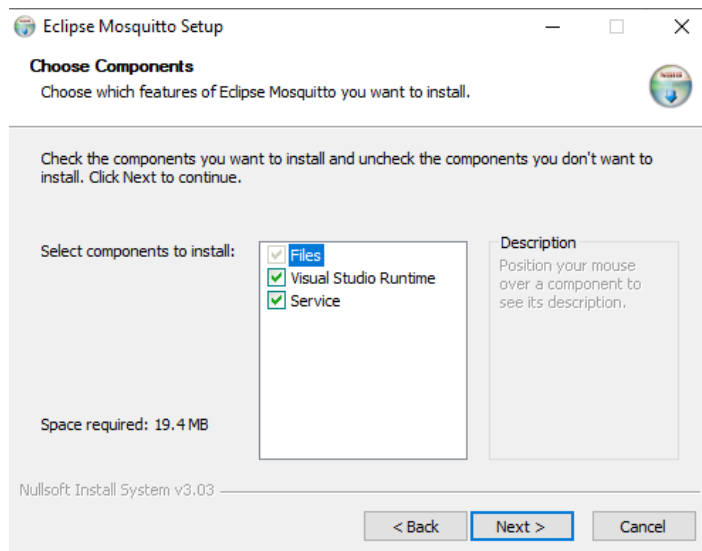
### 4.1 Install Mosquitto

1. Download the required software here:

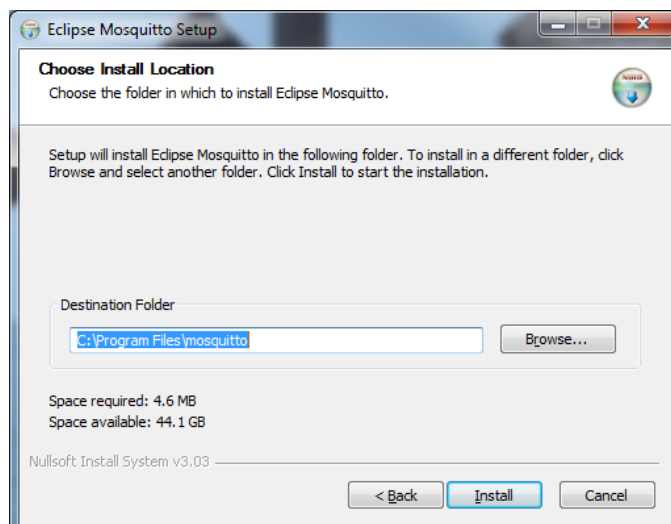
<https://mosquitto.org/download>

2. Double clicking the downloaded file will lead you through the installation of Mosquitto

The next step would be asking if you want to install the service too, leave it selected and click next



3. Select install location as given below



4. Installation finished

The following chapters 4.2 and 4.3 explain how to Start Mosquitto without encryption and how to Start Mosquitto with encryption.

## 4.2 Start Mosquitto without encryption

1. Navigate to the installation folder: C:\Program Files\mosquitto
2. Open the file mosquitto.conf with any text editor like notepad++
3. In the section **Listeners** add a listener on port **1883**
4. In the section **Security**. Uncomment **allow\_anonymous** and set it to **true**
5. Save the file (admin rights required)

```
208
209 # =====
210 # Listeners
211 # =====
212
213 # Listen on a port/ip address combination. By using this variable
214 # multiple times, mosquitto can listen on more than one port. If
215 # this variable is used and neither bind_address nor port given,
216 # then the default listener will not be started.
217 # The port number to listen on must be given. Optionally, an ip
218 # address or host name may be supplied as a second argument. In
219 # this case, mosquitto will attempt to bind the listener to that
220 # address and so restrict access to the associated network and
221 # interface. By default, mosquitto will listen on all interfaces.
222 # Note that for a websockets listener it is not possible to bind to a host
223 # name.
224 #
225 # On systems that support Unix Domain Sockets, it is also possible
226 # to create a # Unix socket rather than opening a TCP socket. In
227 # this case, the port number should be set to 0 and a unix socket
228 # path must be provided, e.g.
229 # listener 0 /tmp/mosquitto.sock
230 #
231 # listener port number [ip address/host name/unix socket path]
232 listener 1883
233
234 # By default, a listener will attempt to listen on all supported IP protocol
235 # versions. If you do not have an IPv4 or IPv6 interface you may wish to
236 # disable support for either of those protocol versions. In particular, note
237 # that due to the limitations of the websockets library, it will only ever
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310 # =====
311 # Security
312 # =====
313
314 # If set, only clients that have a matching prefix on their
315 # clientid will be allowed to connect to the broker. By default,
316 # all clients may connect.
317 # For example, setting "secure-" here would mean a client "secure-
318 # client" could connect but another with clientid "mqtt" couldn't.
319 #clientid_prefixes
320
321 # Boolean value that determines whether clients that connect
322 # without providing a username are allowed to connect. If set to
323 # false then a password file should be created (see the
324 # password_file option) to control authenticated client access.
325 #
326 # Defaults to false, unless there are no listeners defined in the configuration
327 # file, in which case it is set to true, but connections are only allowed from
328 # the local machine.
329 allow_anonymous true
330
```



- To start the broker manually open a command prompt (**CMD**).

#### **No Encryption path:**

Navigate to: `cd C:\Program Files\mosquitto`  
 Call: `mosquitto.exe -c mosquitto.conf -v`



#### **Note:**

The parameter `-c mosquitto.conf` links to the right configuration file  
 The parameter `-v` is optional and is activating the logging

```
Microsoft Windows [Version 10.0.19044.1586]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Test>cd c:/program files/mosquitto

c:\Program Files\mosquitto>mosquitto.exe -c "mosquitto.conf" -v
1647951796: mosquitto version 2.0.14 starting
1647951796: Config loaded from mosquitto.conf.
1647951796: Opening ipv6 listen socket on port 1883.
1647951796: Opening ipv4 listen socket on port 1883.
1647951796: mosquitto version 2.0.14 running
```

The commands above can also be included inside a batch file which can be used to start the Mosquitto broker without encryption. The content of the batch can be copied from below.

```
run cmd.exe
cd "C:\Program Files\Mosquitto"
mosquitto.exe -c mosquitto.conf -v
pause
```

## 4.3 Start Mosquitto with encryption

### 4.3.1 Create self-signed certificates



#### CAUTION!

Self-signed certificates like created and used in this chapter, can be used for test purposes. It's not recommended to use such certificates in a real plant. There certificates signed from an official CA should be used.

Further details about cyber security can be found in our:

- [Whitepaper: Cyber Security in the AC500 PLC](#)
- [AC500 Cyber Security FAQs](#)

If already signed certificates for the mosquitto broker are available, this chapter can be skipped. The configuration of mosquitto is explained in chapter 4.3.2 Adapt mosquito configuration.

If encryption is required, some further steps must be done:

1. Create a **CA key pair**
2. Create **CA certificate** and sign it with the private key from step 1
3. Create the **broker key pair**
4. Create a CA certificate **sign request** using the key from step 3
5. Use the CA certificate from step 2 to **sign the request** from step 4

#### Client Requirements

- A CA (certificate authority) certificate of the CA that has signed the server certificate on the Mosquitto Broker

#### Broker Requirements

- CA certificate of the CA that has signed the server certificate on the Mosquitto Broker
- Server certificate, signed by CA with its private key
- Server Private Key for decrypting of the incoming messages

Before starting, it's recommended to create a new folder. Here a folder **C:\CERT** was created.

For the certificate creation and signing OpenSSL is required. Please visit <https://slproweb.com/products/Win32OpenSSL.html> and download and install the open SSL Light version.

Open the installation folder and C:\Program Files\OpenSSL-Win64 and run the script **start.bat as administrator**.

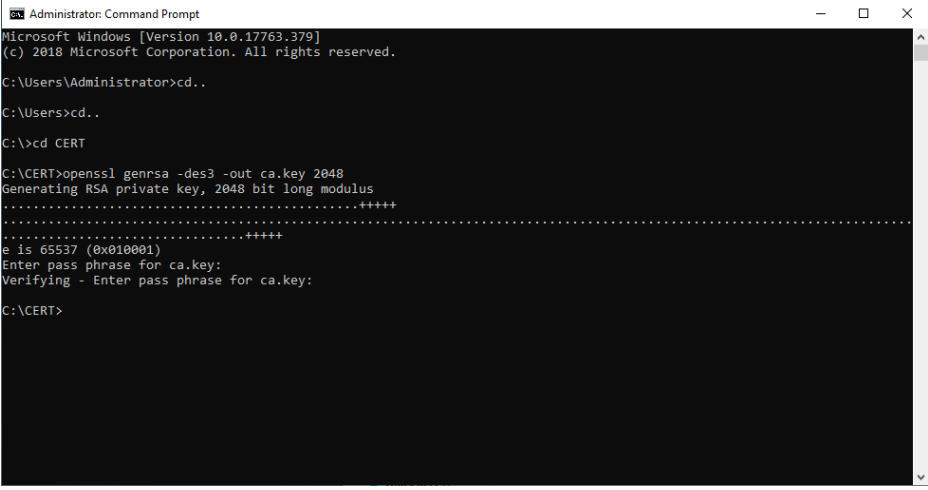
Navigate to the created folder with the command `cd C:\CERT`

If the folder was named different or is on a different location this path needs to be adapted.

## 1. Create CA Key Pair

Run the following command in your CMD

```
openssl genrsa -des3 -out ca.key 2048
```



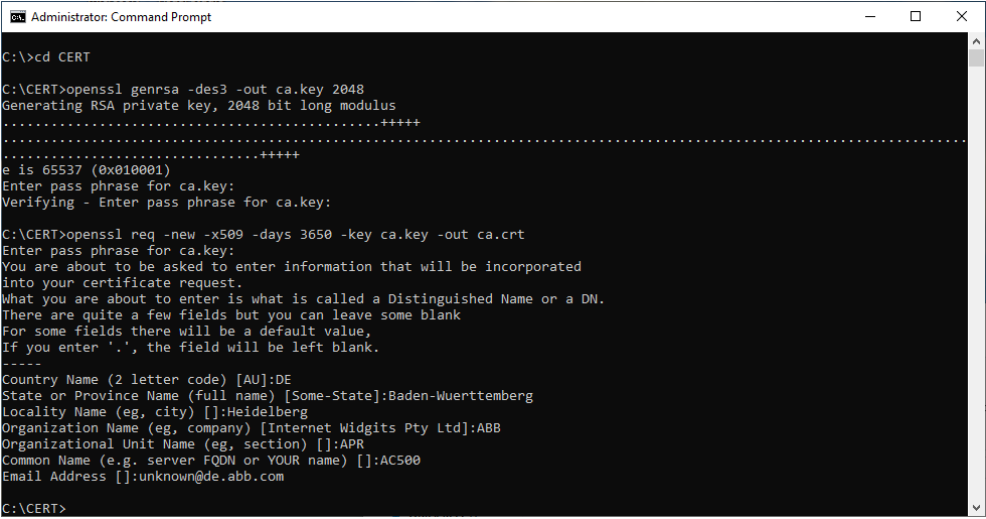
```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.17763.379]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd..
C:\Users>cd..
C:\>cd CERT
C:\CERT>openssl genrsa -des3 -out ca.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++++
.....+++++
e is 65537 (0x010001)
Enter pass phrase for ca.key:
Verifying - Enter pass phrase for ca.key:
C:\CERT>
```

- **genrsa**: generate a RSA private key
- **-des3**: Using DES3 cipher for the key generation
- **-out**: specifies the output file name (.key)
- **2048**: number of bits for the private key

## 2. Create CA Certificate

```
openssl req -new -x509 -days 3650 -key ca.key -out ca.crt
```

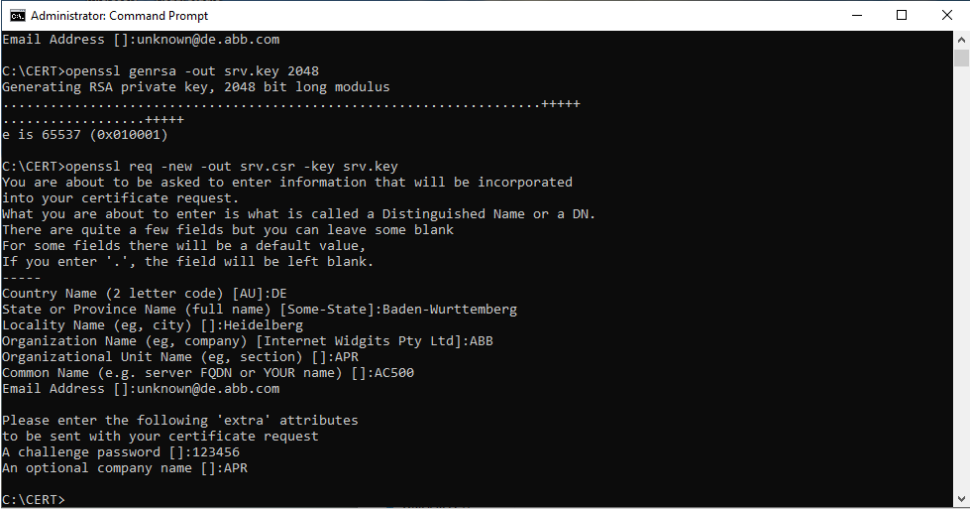


```
Administrator: Command Prompt
C:\>cd CERT
C:\CERT>openssl genrsa -des3 -out ca.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++++
.....+++++
e is 65537 (0x010001)
Enter pass phrase for ca.key:
Verifying - Enter pass phrase for ca.key:
C:\CERT>openssl req -new -x509 -days 3650 -key ca.key -out ca.crt
Enter pass phrase for ca.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:Baden-Wuerttemberg
Locality Name (eg, city) []:Heidelberg
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ABB
Organizational Unit Name (eg, section) []:APR
Common Name (e.g. server FQDN or YOUR name) []:AC500
Email Address []:unknown@de.abb.com
C:\CERT>
```

- **req**: certificate request and certification utility.
- **-new**: generate new certificate, it will prompt user for several input fields.
- **-x509**: create a self-signed certificate.
- **-days**: specify the number of days the certificate is valid.
- **-key**: key file with private key to be used for signing
- **-out**: specifies the file name for the certificate (.crt)

### 3. Create **Mosquitto Broker Key Pair**

```
openssl genrsa -out srv.key 2048
```



```
Administrator: Command Prompt
Email Address []:unknown@de.abb.com

C:\CERT>openssl genrsa -out srv.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++++
.....+++++
e is 65537 (0x010001)

C:\CERT>openssl req -new -out srv.csr -key srv.key
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:Baden-Wuerttemberg
Locality Name (eg, city) []:Heidelberg
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ABB
Organizational Unit Name (eg, section) []:APR
Common Name (e.g. server FQDN or YOUR name) []:ACS00
Email Address []:unknown@de.abb.com

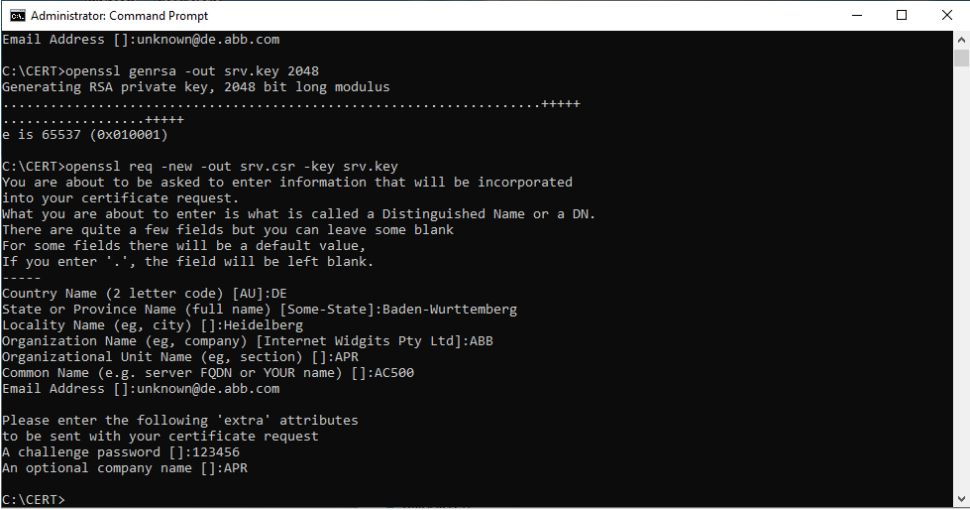
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:123456
An optional company name []:APR

C:\CERT>
```

- **genrsa**: generate a RSA private key
- **-out**: specifies the output file name (.key)
- **2048**: number of bits for the private key

### 4. Create **Certificate Request** from CA

```
openssl req -new -out srv.csr -key srv.key
```



```
Administrator: Command Prompt
Email Address []:unknown@de.abb.com

C:\CERT>openssl genrsa -out srv.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++++
.....+++++
e is 65537 (0x010001)

C:\CERT>openssl req -new -out srv.csr -key srv.key
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:Baden-Wuerttemberg
Locality Name (eg, city) []:Heidelberg
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ABB
Organizational Unit Name (eg, section) []:APR
Common Name (e.g. server FQDN or YOUR name) []:ACS00
Email Address []:unknown@de.abb.com

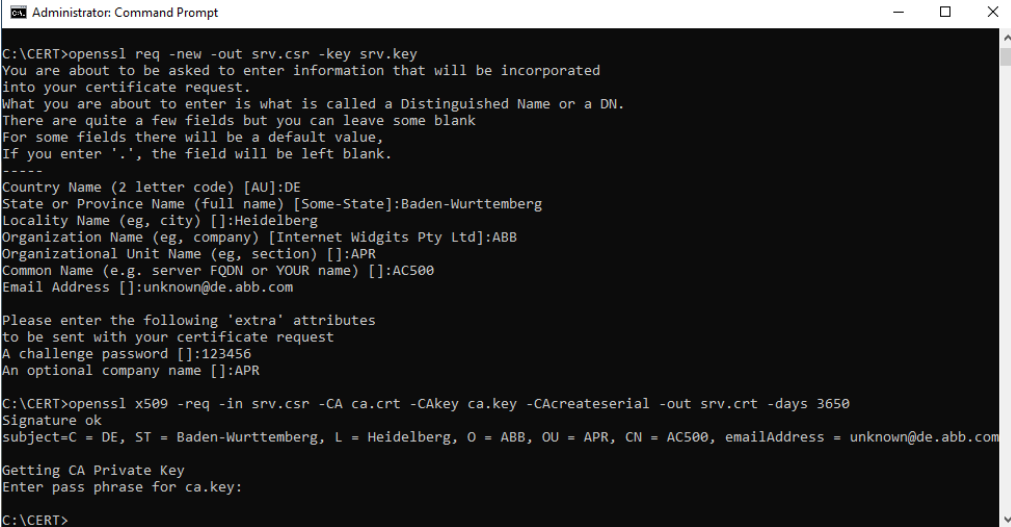
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:123456
An optional company name []:APR

C:\CERT>
```

- **req**: certificate request and certification utility.
- **-new**: create new request file
- **-out**: file name for the certificate signing request (.csr)
- **-key**: file name of the key to be certified.

## 5. Verify and Sign the Certificate Request

```
openssl x509 -req -in srv.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out srv.crt -days 3650
```



```
Administrator: Command Prompt
C:\CERT>openssl req -new -out srv.csr -key srv.key
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:Baden-Wuerttemberg
Locality Name (eg, city) []:Heidelberg
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ABB
Organizational Unit Name (eg, section) []:APR
Common Name (e.g. server FQDN or YOUR name) []:AC500
Email Address []:unknown@de.abb.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:123456
An optional company name []:APR

C:\CERT>openssl x509 -req -in srv.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out srv.crt -days 3650
Signature ok
subject=C = DE, ST = Baden-Wuerttemberg, L = Heidelberg, O = ABB, OU = APR, CN = AC500, emailAddress = unknown@de.abb.com

Getting CA Private Key
Enter pass phrase for ca.key:

C:\CERT>
```

- **x509**: certificate display and signing utility
- **-req**: a certificate request is expected as input
- **-in**: input file for the certificate
- **-CA**: specifies the file to be signed
- **-CAkey**: CA private key to sign the certificate with
- **-CAcreateserial**: the serial number file gets created if it does not exist
- **-out**: output file name
- **-days**: how long the certificate shall be valid

## 6. Created files:

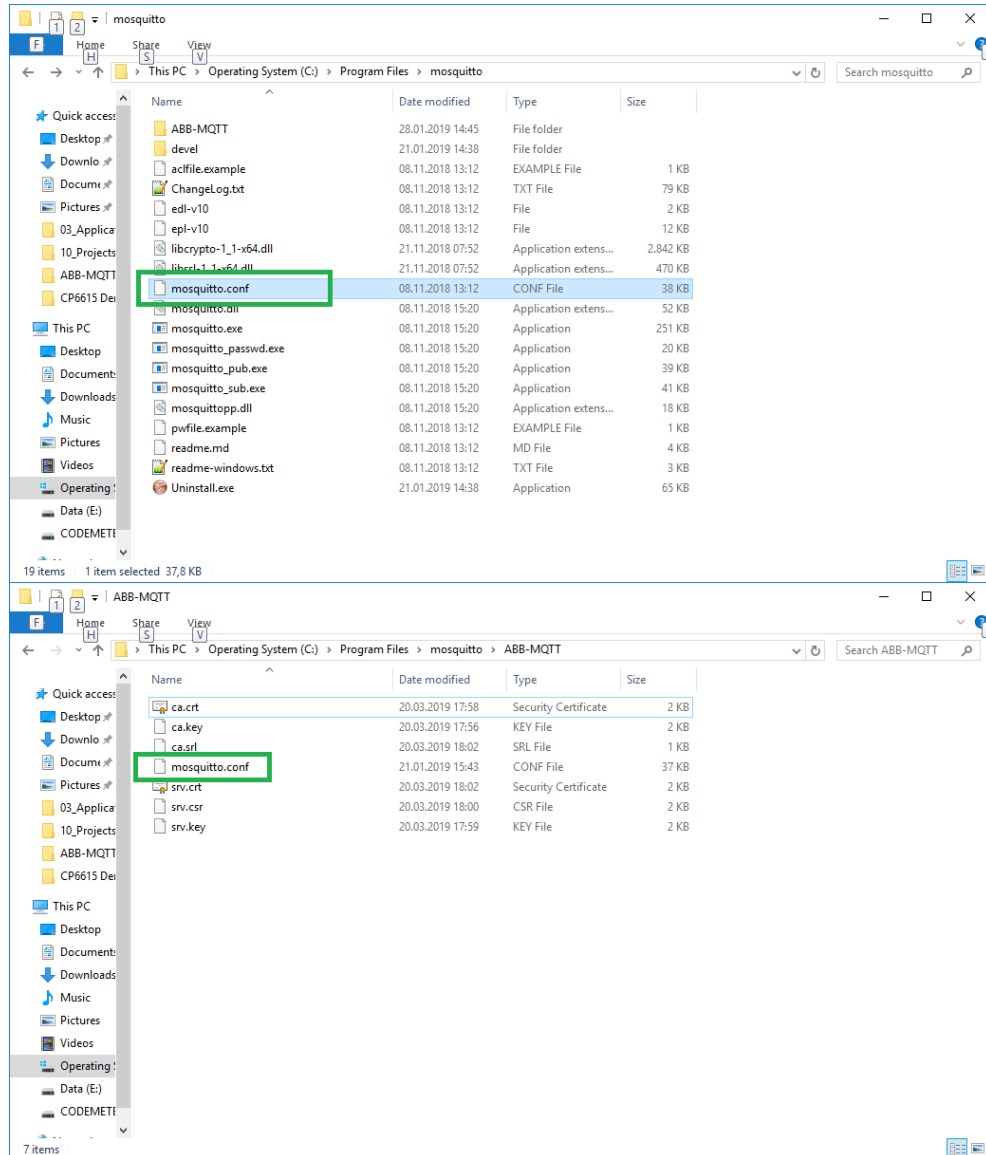
<b>ca.crt</b>	CA Certificate
<b>ca.key</b>	CA key pair (private, public)
<b>ca.srl</b>	CA serial number file
<b>srv.crt</b>	server certificate
<b>srv.csr</b>	certificate sign request, not needed any more
<b>srv.key</b>	server key pair

## 7. Create a folder called: **ABB-MQTT** and copy your generated files from above to the following destination:

**C:\Program Files\mosquitto\ABB-MQTT\**

### 4.3.2 Adapt mosquitto configuration

1. Create a copy of the **mosquitto.conf** and paste it to the **ABB-MQTT** folder:  
This has the advantage that two different configurations for the mosquitto broker are existing. One configuration for a not encrypted and one for an encrypted communication.



2. Adapt the **mosquitto.conf** file inside the **ABB-MQTT** directory. The following lines must be adapted:

The Security setting allow anonymous was already changed in the last chapter it

- needs to be set to **true**. See chapter 4.2

## Listening port

- needs to be changed from 1883 to **8883**. For details see chapter 4.2

## Path to the PEM encoded server certificate

- certfile C:\Program Files\mosquitto\ABB-MQTT\srv.crt

## Path to the PEM encoded keyfile.

- keyfile C:\Program Files\mosquitto\ABB-MQTT\srv.key

## Path to the ca.crt file

- cafile C:\Program Files\mosquitto\ABB-MQTT\ca.crt

```

310
311 # Both of certfile and keyfile must be defined to enable certificate based
312 # TLS encryption.
313
314 # Path to the PEM encoded server certificate.
315 certfile C:\Program Files\mosquitto\ABB-MQTT\srv.crt
316
317 # Path to the PEM encoded keyfile.
318 keyfile C:\Program Files\mosquitto\ABB-MQTT\srv.key
319
320
321
322
323
324
325
326
327
328
329
330
331 # cafile and capath define methods of accessing the PEM encoded
332 # Certificate Authority certificates that will be considered trusted when
333 # checking incoming client certificates.
334 # cafile defines the path to a file containing the CA certificates.
335 # capath defines a directory that will be searched for files
336 # containing the CA certificates. For capath to work correctly, the
337 # certificate files must have ".crt" as the file ending and you must run
338 # "openssl rehash <path to capath>" each time you add/remove a certificate.
339 cafile C:\Program Files\mosquitto\ABB-MQTT\ca.crt
340 #capath
341

```

### 4.3.3 Start Mosquitto broker

Configuration is done. Now we can start the broker. To start the broker manually open a command prompt (CMD).

#### With Encryption path:

Navigate to: `cd C:\Program Files\mosquitto`

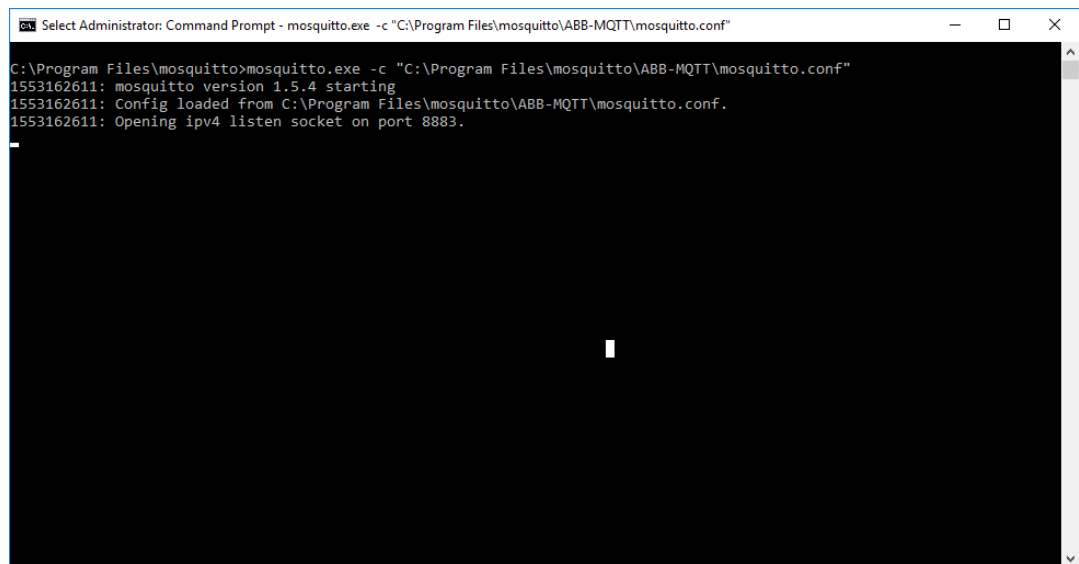
Call: `mosquitto.exe -c "C:\Program Files\mosquitto\ABB-MQTT\mosquitto.conf" -v`



#### Note:

The parameter `-c "C:\Program Files\mosquitto\ABB-MQTT\mosquitto.conf"` links to the right configuration file

The parameter `-v` is optional and is activating the logging



The commands above can also be included inside a batch file which can be used to start the Mosquitto broker with encryption. The content of the batch can be copied from below.

```
run cmd.exe
cd "C:\Program Files\Mosquitto"
mosquitto.exe -c "C:\Program Files\Mosquitto\ABB-MQTT\mosquitto.conf" -v
pause
```



## 5 Mqtt.fx setup & configuration

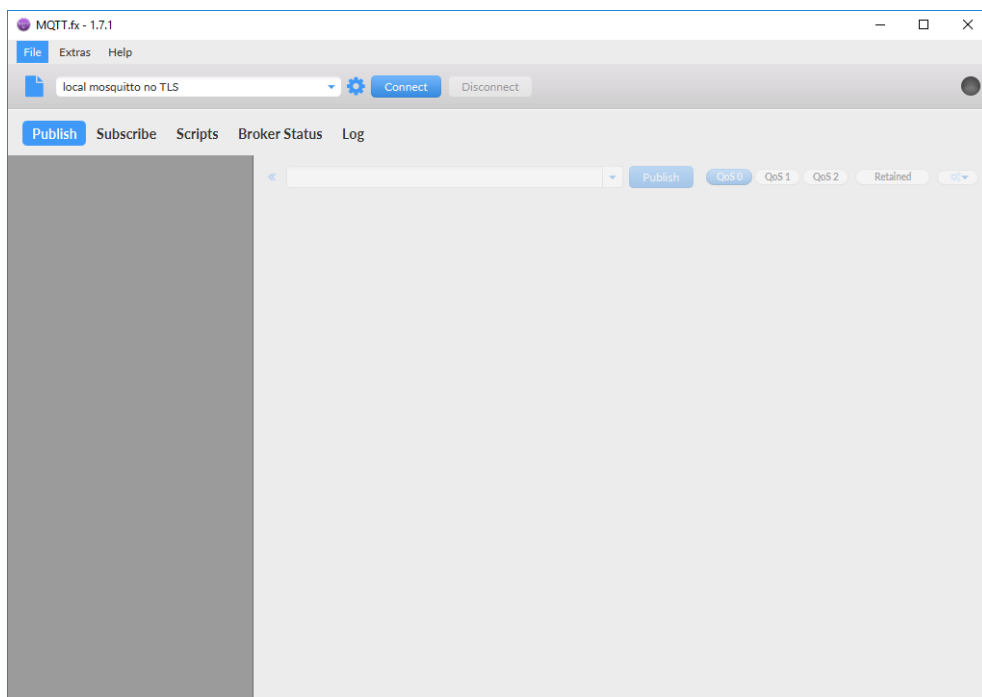
Mqtt.fx is a graphical MQTT client tool. It supports to publish a message or subscribe to a topic.

### 5.1 Install Mqtt.fx

1. Download the required software here:  
<http://www.jensd.de/apps/mqttfx/1.7.1/mqttfx-1.7.1-windows-x64.exe>
2. Follow the instructions in the setup

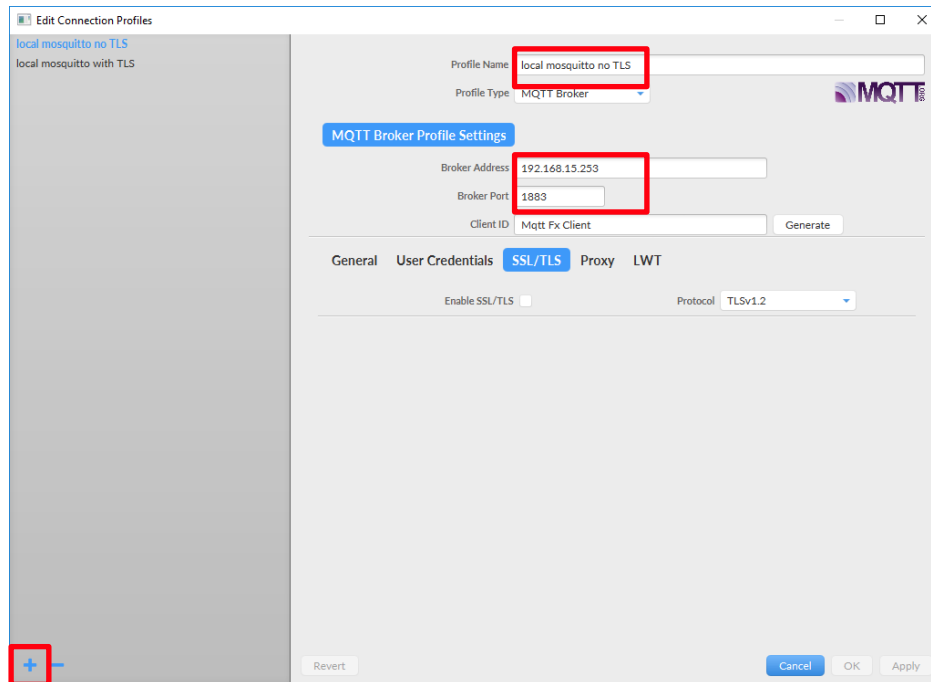
### 5.2 Start Mqtt.fx without encryption

1. After the Mqtt.fx software is started, you will see this screen:

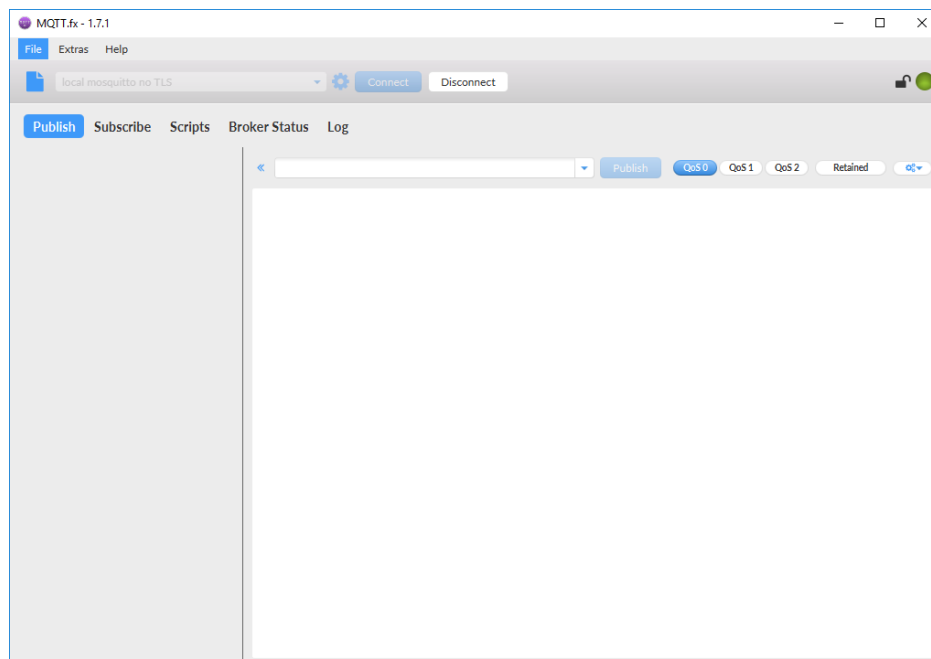


2. To configure a new profile, please click on the icon, left to the **Connect** button. You will see this screen. On the left lower corner, you can click on the **+ sign** to create a new profile.

Here a new profile with name “**local mosquitto no TLS**” was created.



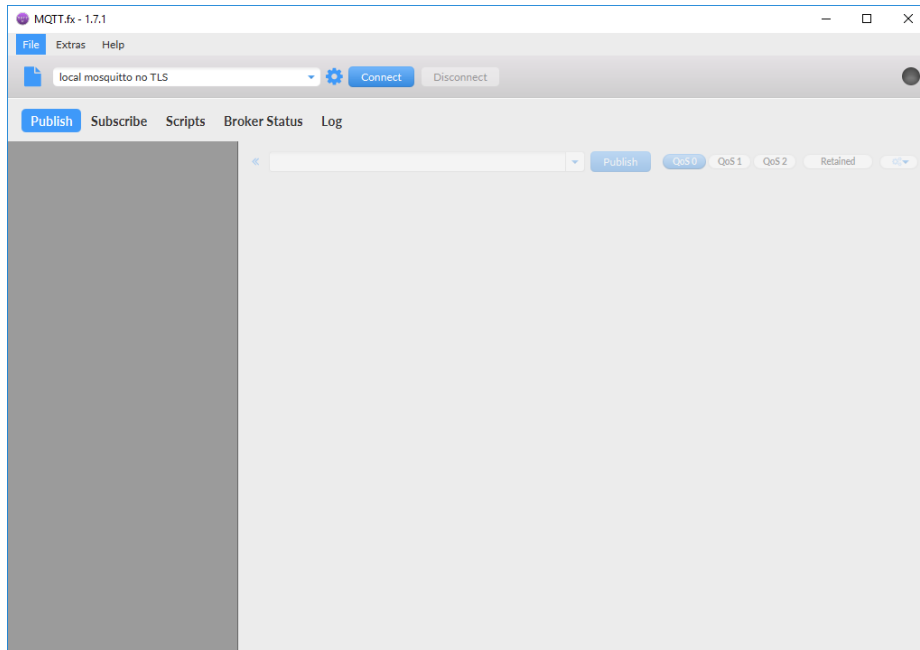
3. Apply your settings and return to the main window.
4. Now, we can connect the **Mqtt.fx client** to the **mosquitto broker**



The successfully connection will be displayed with the **green light** in the right upper corner. The lock is open, this means **no secure communication** is used.

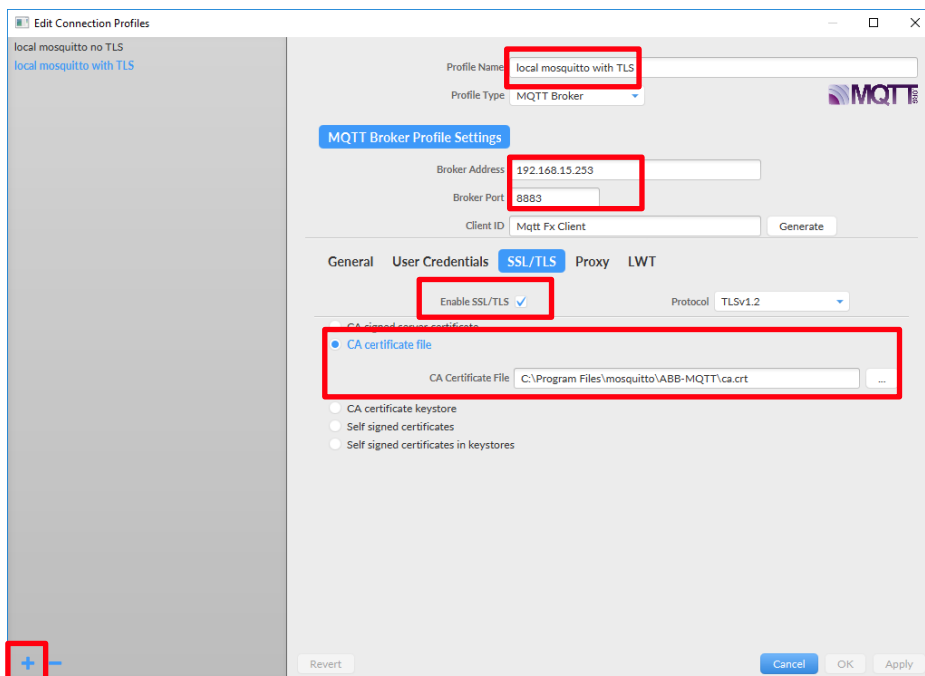
## 5.3 Start Mqtt.fx with encryption

1. After the Mqtt.fx software is started, you will see this screen:



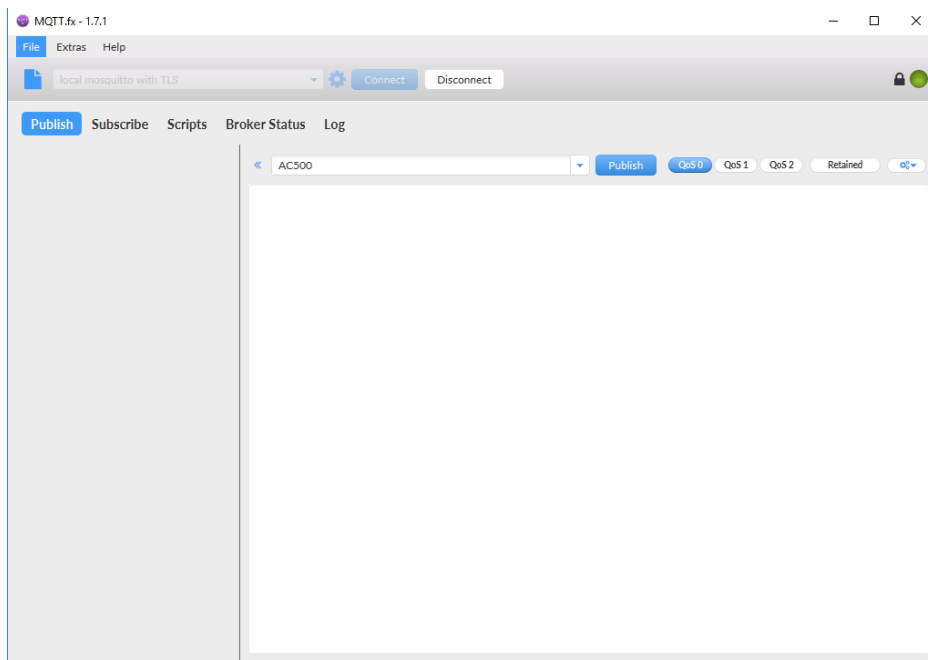
2. To configure a new profile, please click on the icon, left to the **Connect** button. You will see this screen. On the left lower corner, you can click on the **+ sign** to create a new profile.

Here a new profile with name “**local mosquitto with TLS**” can be created



Very important on the picture above is, that the **SSL/TLS** tab is connected to the **ca.crt** file, located on the ABB-MQTT folder.

3. Apply your settings and return to the main window.
4. Now, we can connect the **Mqtt.fx client** to the **mosquitto broker**



The successfully connection will be displayed with the **green light** in the right upper corner. The lock is closed, this means **secure communication** is used.

## 6 PLC Configuration

### 6.1 Settings to establish cloud connection

After successful setup of the above mentioned mosquitto broker, we can set the inputs within the MQTT-library inside the PLC application.

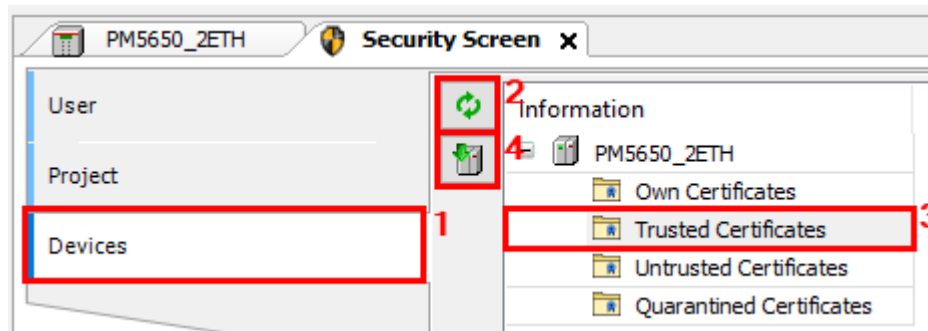
Please check the list MqttVars in the folder 01\_Vars

The variables in this list are used for the MQTT function blocks

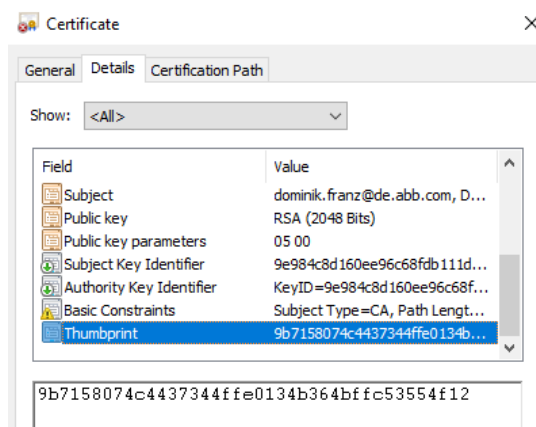
Please change ServerPort, ServerName, PathToCertStore and ToggleSecurity according to your needs.

If you're working with encrypted communication the "ca.crt" needs to be available in the PLC. You can load it into the certificate store of the PLC by opening View > Security screen

Open the tab "Devices"(1) and refresh(2) to find your PLC. The gateway needs to be connected. Click "Trusted Certificates"(3) and download (4) the certificate to the PLC.



The certificate will appear on the right side. Double click on the certificate and open the tab "Details". Scroll down to find at the end of the list the "Thumbprint" which is required for the Mqtt connect as "ServerCert". Copy this Thumbprint



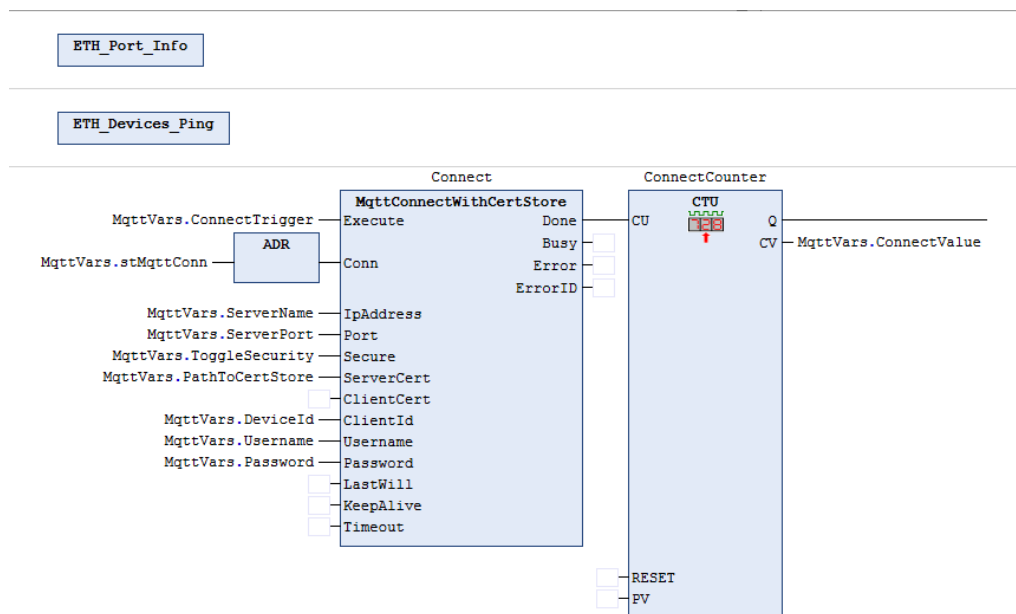
Each certificate has a validation time. Please make sure, that the time inside the PLC is set.

This application example has the same functionality on the one hand as FBD program and on the other hand as ST program.

Depending on what you like more you can change the call in NewTask from MosquittoSt to MosquittoFbd. Please make sure that not both programs are called the same time.

Most important for the first communication is the connect to the broker. The connect function block has several inputs, which are set by the global Variables in MqttVars

- Con: Connection Pointer (must be connected to each Mqtt FB)
- IpAddress: 192.168.15.253 (IP address of the computer where mosquitto is running)
- Port: 8883 (default port for TLS connection)
- Secure: True (We want to enable security)
- ServerCert: We are using the Thumbprint of “ca.crt”
- ClientCert: Not required
- ClientKey: Not required
- ClientID: AC500
- Username: Not required
- Password: Not required
- LastWill: Testament can be added here
- KeepAlive: 600s (default)
- Timeout: 30s (default)



## 6.2 Settings for publishing & subscribing messages

MQTT can publish messages to different topics in contrast to Azure for Mosquitto any topic can be used for **publishing & subscribing** messages:

For example, we are using following topic:

**AC500**

This topic can be used for **publishing & subscribing**.

## 7 FAQs

### What can I do if I cannot connect to my mosquitto broker with my AC500?

Please activate the logging inside the mosquitto broker by adding “-v” to the call as described in the chapters above.

### What can I do if Mosquitto gives the warning: Starting in local only mode. Connections will only be possible from clients running on this machine.

Please make sure that the right config file is loaded, and the setting allow anonymous is set to true.

### What can I do if Mosquitto gives the error: Only one usage of each socket address (protocol/network address/port) is normally permitted.

There is already the mosquitto service running. Please either quit existing CMD windows, where mosquitto was started or stop the service “Mosquitto” in the task manager in the tab “Services”.

### I’ve done all settings as described above. A communication with Mqtt.fx is working but not with an AC500. The ErrorId in the AC500 says only Timeout and I cannot see anything in the Mosquitto logging

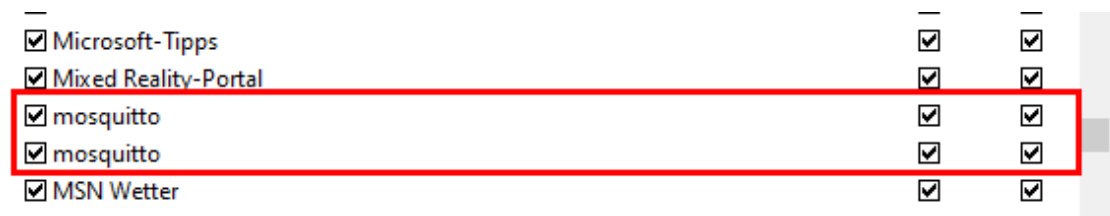
Please check your firewall settings

Settings > Update & Security > Windows Security > Open Windows Security

Firewall & network protection > Allow an app through firewall

Click Change settings with administrator rights

Scroll down to mosquitto and allow this app to pass the firewall



For further questions please check our Application Note **AC500 MQTT FAQs**





---

ABB AG  
Eppelheimer Straße 82  
69123 Heidelberg, Germany  
Phone: +49 62 21 701 1444  
Fax: +49 62 21 701 1382  
E-Mail: [plc.support@de.abb.com](mailto:plc.support@de.abb.com)  
[www.abb.com/plc](http://www.abb.com/plc)

---

We reserve the right to make technical changes or modify the contents of this document without prior notice. With regard to purchase orders, the agreed particulars shall prevail. ABB AG does not accept any responsibility whatsoever for potential errors or possible lack of information in this document.

We reserve all rights in this document and in the subject matter and illustrations contained therein. Any reproduction, disclosure to third parties or utilization of its contents – in whole or in parts – is forbidden without prior written consent of ABB AG.  
Copyright© 2022 ABB. All rights reserved