AWS IoT Core Client SL

The “AWS IoT Core Client SL” library allows for the exchange of messages via the “AWS IoT Core” Amazon Service.

Product description

The “AWS IoT Core” service is a managing cloud platform from Amazon in which connected devices can work together easily and safely with cloud applications and other devices. The “AWS IoT Core Client SL” library provides function blocks for sending and receiving messages. Communication is encrypted and takes place by means of the MQTT protocol. In the AES environment, messages are typically transmitted in JSON format. The “JSON Utilities” library can be used for parsing and creating JSON files.

The following function blocks are included in the library:

- `AWSIoTClient`: Function block for establishing a connection to “AWS IoT Core”
- `AWSIoTPublish`: Function block for publishing messages (Publish)
- `AWSIoTSubscribe`: Function block for subscribing to a topic (Subscribe)
- `AWSIoTGetDeviceShadow`: Function block for reading the “Device Shadow”
- `AWSIoTUpdateDeviceShadow`: Function block for refreshing the “Device Shadow” data
- `AWSIoTSubscribeDeviceShadow`: Function block for subscribing to “Device Shadow” changes

Supported functions

- Publishing and subscription of messages based on MQTT V3.1.1
- TLS encryption
- Authentication via client certificate
- Support of “Quality of Service”: 0 and 1 (QoS0, QoS1)
- Data type “Topics”: `WSTRING`
- Maximum size of a topic: 1024
- The maximum package size and payload size can be configured by means of a parameter list.
- Multitasking and multicore support
- Support of “Last Will” messages (QoS0, QoS1)
- Support of wildcards (# and +)

The sample project “AWS IoT Core Client SL Example.project” is installed in the target directory. The application `AWSDeviceShadow` demonstrates the reading and writing of the “Device Shadow”. The application `AWSPubSub` demonstrates how messages can be published and

subscribed to by means of the included function blocks. The application `AWS_JSON_DeviceShadow` demonstrates how a device shadow can be updated via the JSONBuilder functionality.

The MQTT parameters can be changed by adding the library “MQTT Client SL” on toplevel in the Library Manager.

Installation of a client certificate

Devices for “AWS IoT Core” are authenticated by means of client certificates. The client certificate can be created and downloaded with a Certificate Signing Request (CSR) in the AWS Management Console. Then the certificate can be installed on the corresponding device.

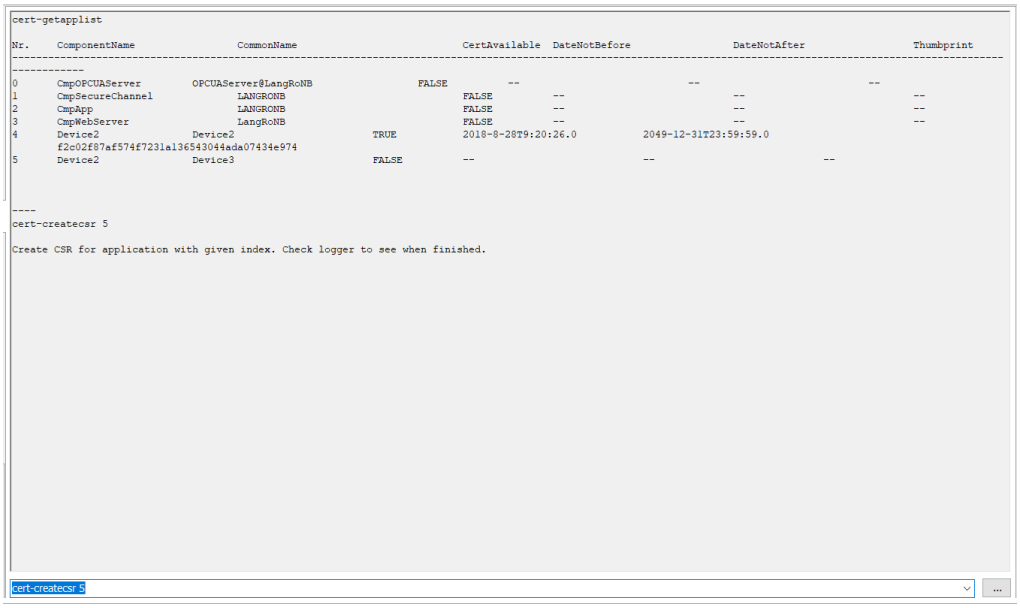
To perform the subsequent steps, the CODESYS package “CODESYS Security Agent” and “OpenSSL” must be installed on the PC.

Download links:

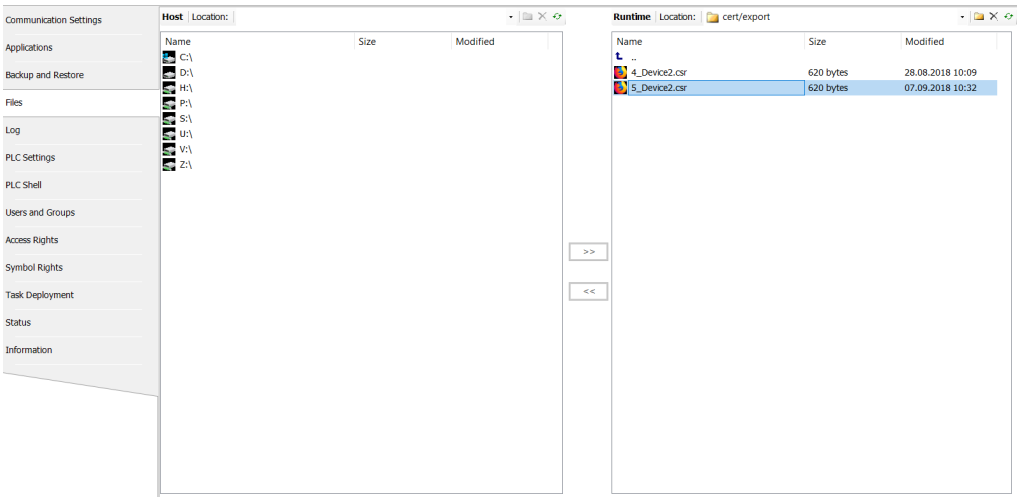
- CODESYS Security Agent: https://store.codesys.com/codesys-security-agent.html
- OpenSSL: https://www.openssl.org

Steps for creating and importing a client certificate:

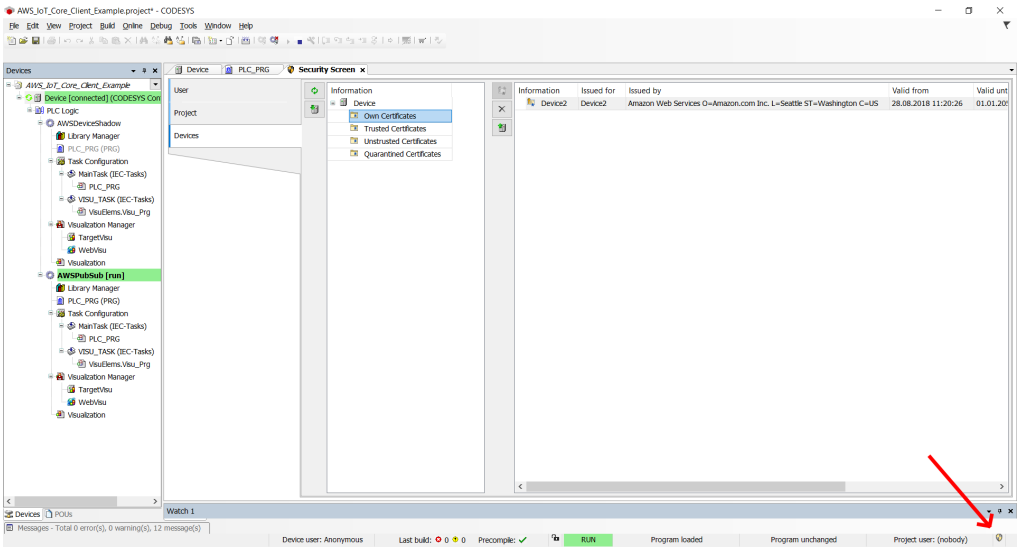
1. Create a device in “AWS IoT Core” (see https://docs.aws.amazon.com/iot/latest/developerguide/register-device.html).
2. Open the application `AWSPubSub` in the sample project.
3. Set the end point to the input `AWSIoTClient.sHostname` (for example, xxxxxxxxxxxxxx.iot.yyyyyyyy.amazonaws.com).
4. Set the device name (name of thing) to the input `AWSIoTClient.sClient`. The input `AWSIoTClient.sCertCNPrefix` should be used if the ClientId is identical to the host name. This prevents loading of a wrong certificate (e.g. from the WebServer). Example: `sCertCNPr := 'AWSClient'`, `sClientId := 'MyHostname'` => The Common Name (CN) of the certificate will be AWSClient@MyHostname
5. Download and start the project.
6. Open the PLC shell (Device -> PLC-Shell).
7. Specify `cert-getapplist`. -> A component with the specified device name and a `number` is displayed.
8. Specify `cert-createcsr <number>` and use the `number` from step 7. The creation of the CSR file can take several seconds. A corresponding message is displayed in the device log (Device -> Log) after it has been created.



1. Open (Device -> Files) and copy the CSR file from the `cert/export` directory to the local file system.



1. Currently, the format of the exported CSR file is not yet supported by AWS. Until the problem is solved, the file must be converted via OpenSSL. Execute the following command in order to convert the format:
   `openssl.exe req -in YOUR.csr -inform der -out YOUR.csr`
2. A certificate can now be created in the AWS Management Console (“Create with CSR”) with the `YOUR.crt` file. (see https://docs.aws.amazon.com/iot/latest/developerguide/create-device-certificate.html)
3. Download the certificate and root certificate.
4. Security Screen -> Install the devices and certificate in “Own Certificates”. Install the root certificate in “Trusted Certificates”.



1. Menu bar: Online -> Reset Cold. Start the project.
2. The output `AWSIoTClient.xConnectedToBroker` should be set to TRUE. -> The connection has been established.

General information

Supplier:

CODESYS GmbH
Memminger Strasse 151
87439 Kempten
Germany

Support:

Technical support is not included with this product. To receive technical support, please purchase a CODESYS Support Ticket.

https://support.codesys.com

Item:

AWS IoT Core Client SL

Item number:

Sales/Source of supply:

CODESYS Store
https://store.codesys.com

Included in delivery:

CODESYS package

System requirements and restrictions

Programming system	CODESYS Development System V3.5.15.0 or later
Runtime system	CODESYS Control V3.5.15.0 or later
Supported platforms and devices	Note: Use the “Device Reader” project for locating the functions in the CODESYS Store free of charge.
Additional requirements	AWS Account (AWS IoT Core)
Restrictions	-
Licensing	License activation optional on CODESYS Key or Soft Key (Soft Key: free of charge component of CODESYS Controls) Licensing via Soft Key is strictly linked to hardware. Note: Without a license the software runs for 30 minutes in demo mode.
Required accessories	-

Note: Technical specifications are subject to change. Errors and omissions excepted. The content of the current online version of this document applies.

Creation date: 2023-08-21