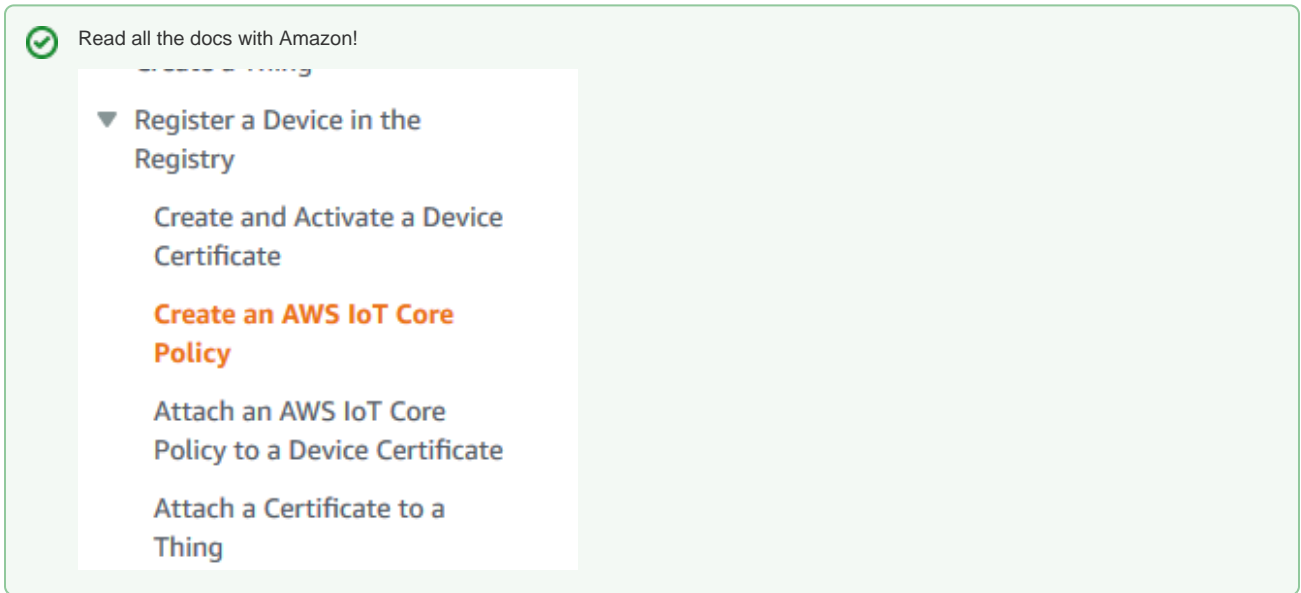


AWS IoT Core Client - getting started

See our CODESYS Store product: [AWS IoT Core Client](#)


How to get started

1. Install the "Security Agent" Addon from our [CODESYS Store](#)
2. Choose from this list, an openssl binary to install:
<https://wiki.openssl.org/index.php/Binaries> - The first entry has worked well in tests
3. Go to aws.amazon.com and sign up




We follow slightly different steps.

4. Click "All Services" and IOT Core
5. Click "Manage"
6. Click "Register a Thing"
7. Click "Create a single Thing" and name it e.g. "**jackicpi3lcd**"
8. Create a type "CODESYS" / "Things connected with CODESYS"

 Don't bother with group, skip all the optional things and hit "next"

9. Create the "Thing" without Certificate, and hit "Done"
10. Open your "Thing" and go to "Interact"
11. Copy the rest API endpoint under HTTPS, e.g. "xxxxxxxxxx-ats.iot.us-east-2.amazonaws.com"
12. Go back to the main menu of AWS IoT, Under Menu "Secure > Policies", press "Create new policy".

 Your account ID is different from the URL in the Rest API mentioned above! It is given as the default Resource when you create a policy.

13. Name it **LetJackPubAndSub**
14. You want to be able to connect with a client ID, publish the topic and the last will, Subscribe to the topic, and then, once subscribed, you also want to receive messages from the topic.
15. Action = **iot:Connect**
16. Resource ARN = arn:aws:iot:<AWS Region>:<AccountID, not same as Rest API URL>:client/**jackicpi3lcd**
17. Effect = Allow
18. Click Add Statement
19. Action = **iot:Publish**

20. Resource ARN = arn:aws:iot:<AWS Region>:<AccountID>, not same as Rest API URL>:topic/hello/aws, arn:aws:iot:<AWS Region>:<AccountID>, not same as Rest API URL>:topic/**jackicpi3lcd**/lastwill
21. Effect = Allow
22. Click Add Statement
23. Action = **iot:Subscribe**
24. Resource ARN = arn:aws:iot:<AWS Region>:<AccountID>, not same as Rest API URL>:topicfilter/hello/aws
25. Effect = Allow
26. Click Add Statement
27. Action = **iot.Receive**
28. Resource ARN = arn:aws:iot:<AWS Region>:<AccountID>, not same as Rest API URL>:topic/hello/aws
29. Effect = Allow
30. Click Create
31. Example:

Version 19 updated Feb 24, 2020 2:56:02 PM +0100

[Edit policy document](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:Connect",
      "Resource": "arn:aws:iot:us-east-2:111111111111:client/jackicpi3lcd"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Publish",
      "Resource": [
        "arn:aws:iot:us-east-2:111111111111:topic/hello/aws",
        "arn:aws:iot:us-east-2:111111111111:topic/jackicpi3lcd/lastwill"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:iot:us-east-2:111111111111:topicfilter/hello/aws"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Receive",
      "Resource": "arn:aws:iot:us-east-2:111111111111:topic/hello/aws"
    }
  ]
}
```

32. CODESYS IDE: Run the installed example from
e.g. `C:\Users\<USERNAME>\AWS IoT Core Client\<VERSION>\AWS_IoT_Core_Client_Example.project`
33. Right click the Device in Device Tree, select "Update Device...", then select your target PLC, before hitting OK



Important step for every example project!

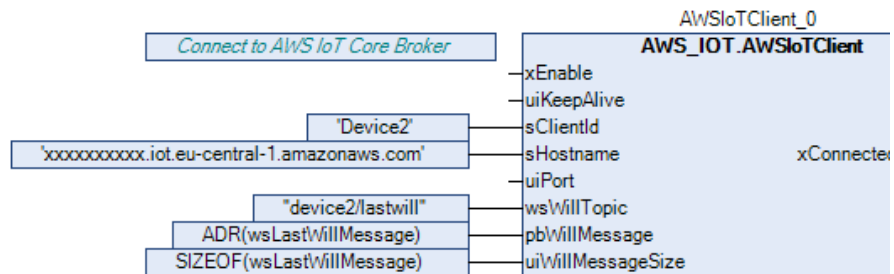
34. Under the Application called AWSPubSub, Open the PLC_PRG and change these settings in the screenshot below to match the previously created "Thing"
 - a. Change the two instances of 'Device2' to '**jackicpi3lcd**'
 - b. Change the Hostname to the one you copied from the interact tab

c. Example before:

```

1  // This example shows how to connect to AWS IoT Core and how to subscri
2  PROGRAM PLC_PRG.
3  VAR.
4  AWSIoTClient_0: AWS_IOT.AWSIoTClient; // Function block to connect
5  AWSIoTPublish_0: AWS_IOT.AWSIoTPublish; // Function block to publish

```



35. Download and run

36. Open the visualization and select "xEnable" for the AWSIoTClient.

⚠ It will NOT show that it is xConnectedToBroker, because we haven't configured the permissions yet.

37. Open the PLC shell:

Device -> [PLC-Shell \(OLH\)](#)

38. Specify **cert-getapplist**

-> A component with the specified device name and a number is displayed.

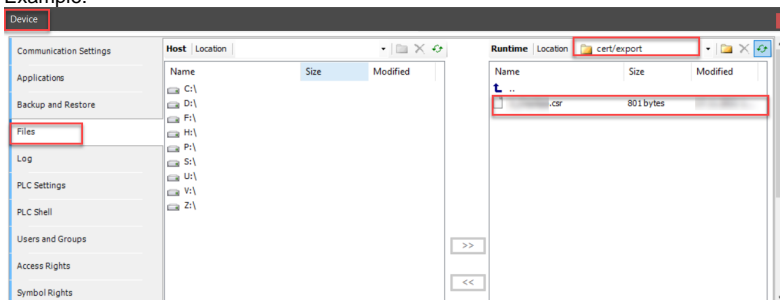
You want the one that says "**jackicpi3lcd**"

39. Specify **cert-createcsr <number>** and use the number from the previous step.

⚠ The creation of the CSR file can take several seconds.
A corresponding message is displayed in the device log (Device -> [Log \(OLH\)](#)) after it has been created.

40. Open (Device -> [Files \(OLH\)](#)) and copy the CSR file from the cert/export directory to the local file system.

41. Example:



42. Open a command prompt and type something like:

```
openssl.exe req -in "D:/prj/WIP/AWS IOT CORE/6_jackicpi3lcd.csr" -inform der -out "D:/prj/WIP/AWS IOT CORE/jackicpi3lcd.csr"
```

43. Go back to the AWS management console

44. Open your "**Thing**" and go to the security console

45. Hit "View other options"


46. Hit "create with csr"

47. Use the output from the above openssl command ("**jackicpi3lcd.csr**")

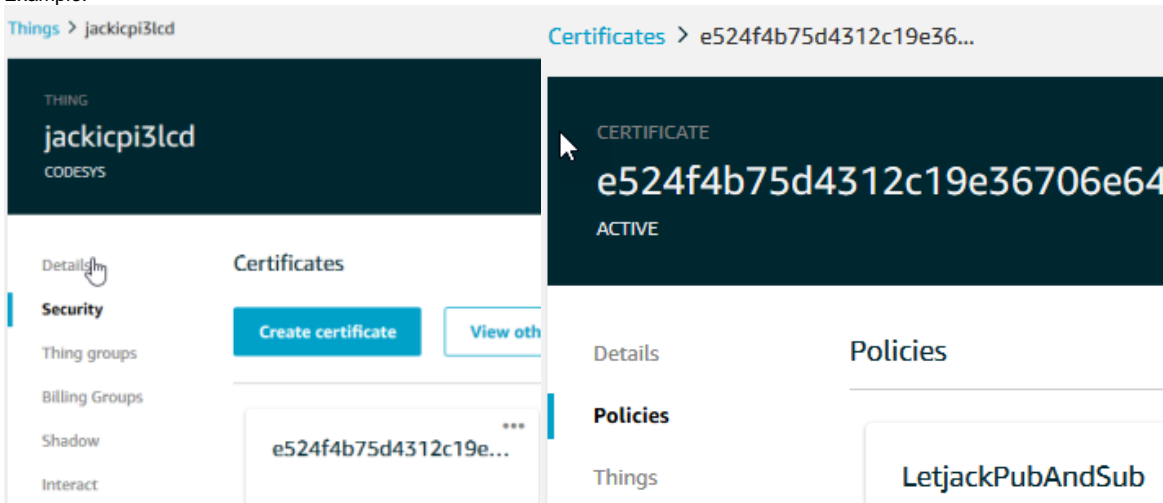
48. Hit "Upload File".

If it doesn't say successful after 10 seconds, you've grabbed the wrong file.


49. Download the certificate it gives you

 Depending on your browser, will actually save as .txt for some reason.

50. Hit the link to download a root certificate.
Grab Amazon root ca1, Root CA3, and startfield Root CA Certificates.
51. Press Activate.
52. Press Attach a policy
53. Select the policy you created earlier
54. Now it won't have added the certificate to the thing for some reason..
55. Go back to the main menu and select Secure > Certificates, select the new one you made.
56. Actions > Activate
57. Actions > Attach Thing > **jackicpi3lcd** > Attach
58. So now if you go to Manage > Thing > **jackicpi3lcd**, it should have a certificate under security.
If you click that it should have a policy under Policies, and the certificate should have the word "**ACTIVE**".
59. Example:



60. Go back to the CODESYS IDE now.
61. [Security screen > Devices](#) > Hit Refresh > Click on "Own Certificates"
62. Press the button just to the left of Owned certificates that looks like a PLC with a green downwards arrow.
63. Example:

 **Download: Transfer the selected certificate to the PLC**

64. Change the file filter to all files.
Change the extension of the file you downloaded from .txt to .crt, then select it and hit open.
65. Go to trusted certificates folder now, and install all three root certificates (same button) from Amazon.
Again, changing the filter if needed.
66. Cold reset the controller to take the new certificates.
67. Open the visualization, turn on **xEnable** again for **AWSIoTClient**.
You should get a green "connected to broker" lamp.
68. Turn on **xEnable** the **AWSIoTSubscribe**
69. Enable the **AWSIoTPublish**
70. Now you should get the values, everything is good!
The topic and payload you send via Publish should be seen in the **PLC_PRG.AWSIoTSubscribe_0**

71. Example:

PubSub Example

Connection Settings

AWSIoTClient

Instance: PLC_PRG AWSIoTClient_0

xEnable

☒

uiKeepAlive

10

sClientId

jackicpi3lcd

sHostname

s.iot.us-east-2.amazonaws.com

uiPort

8883

wsWillTopic

jackicpi3lcd/lastwill

eLastWillQoS

QoS1

xError

☐

xBusy

☒

eError

0

xConnectedToBroker

☒

Subscribe Topic

AWSIoTSubscribe

Instance: PLC_PRG AWSIoTSubscribe_0

xEnable

☒

eQoS

QoS1

xError

☐

xBusy

☒

eError

0

xReceived

☐

udiPayloadSize

81

xSubscribeActive

☒

wsLastTopic

hello/aws

Publish Message

AWSIoTPublish

Instance: PLC_PRG AWSIoTPublish_0

xExecute

☒

udiTimeOut

0

eQoS

QoS1

xError

☐

xBusy

☐

xDone

☒

eError

0
