Amazon Web Service IoT provider

Version 1.2.1

User's guide

January 11, 2023

【 remarks 】

This document uses the machine translation.

【 revision history 】

Version	Date	Content
1.0.0	2017-05-22	First edition.
1.1.0	2021-03-19	Supports TLS 1.2. Change to EXE format.
1.1.1	2021-07-09	Changed OpenSSL to 1.1.1k. Supports Server Name Indication (SNI).
1.2.0	2021-12-28	Supports Websocket. Changed Optional character string of CaoWorkspace::AddController.
1.2.1	2023-01-11	Changed OpenSSL to 3.0.

【 hardware 】

Model	Version	Notes

Contents

1. Introduction 4

2. Outline of provider 5

2.1. Outline 5
2.2. Method property 6
2.2.1. CaoWorkspace::AddController method 6
2.2.2. CaoController::Execute method 7
2.2.3. CaoController::AddVariable method 7
2.2.4. CaoVariable:get_VariableNames property 7
2.2.5. CaoVariable:get_Value property 8
2.2.6. CaoVariable:put_Value property 8
2.2.7. CaoController::OnMessage event 8
2.2.7.1. AWSIoT event reception 8
2.3. Command list 8
2.3.1. CaoController class 8
2.4. Variable list 12
2.4.1. CaoController class 12

1. Introduction

This book is an user's guide of the CAO provider that sends and receives data for AWSIoT of Amazon Web Service.

CAO provider (CaoProvAWSIoT.exe) that treats in this book is called AWSIoT provider.

Capter 2, Details of the outline and the variable of the AWSIoT provider have been described to the chapter.

This provider is using AWS IoT C++ Device SDK and AWS SDK for .NET to use it to communicate with AWSIoT.

Please refer to the following sites for these.

".Net Framework 4.5.2" is necessary to use this provider.

Site links of AWS IoT C++ Device SDK

・AWS IoT C++ Device SDK

https://github.com/aws/aws-iot-device-sdk-cpp

Copyright and license of AWS IoT C++ Device SDK

This application program contains the product distributed by the license of Apache License and Version 2.0.

https://github.com/aws/aws-iot-device-sdk-cpp/blob/master/LICENSE

Site links of AWS SDK for .NET

・AWS SDK for .NET

https://github.com/aws/aws-sdk-net/

Copyright and license of AWS SDK for .NET

This application program contains the product distributed by the license of Apache License and Version 2.0.

https://github.com/aws/aws-sdk-net/blob/master/License.txt

2. Outline of provider

2.1. Outline

The AWSIoT provider is CAO provider that communicates with AWSIoT by using AWS IoT C++ Device SDK and AWS SDK for .NET. The file format is EXE, and when using it from the CAO engine, it is dynamically loaded. To use the AWSIoT providerTable2-1It is necessary to register drinking by the method. RegistAsm.bat and UnregistAsm.bat are in the DotNet¥BAT folder under the folder that installs ORiN2SDK.

Table2-1AWSIoT provider

File name	CaoProvAmazonAWSIoT.exe
ProgID	CaoProv.Amazon.AWSIoT
Registry registration	RegistAsm.bat CaoProvAmazonAWSIoT.exe
Blotting out of registry registration	UnregistAsm.bat CaoProvAmazonAWSIoT.exe

2.2. Method property

2.2.1. CaoWorkspace::AddController method

The AWSIoT provider connects it with AWSIoT referring to connected parameter for the communication at AddController.

Format AddController(<bstrCtrlName:BSTR>,<bstrProvName:BSTR>,
<bstrPCName:BSTR>,<bstrOption:BSTR>))

bstrCtrlName : In controller name
bstrProvName : In provider name. " CaoProv.Amazon.AWSIoT fixed value ="
bstrPcName : Execution machine name of in provider
bstrOption : In optional character string

Table2-2Optional character string of CaoWorkspace::AddController

[1]Option	Explanation
Endpoint=<AWSIoT EndPoint>	Required. Endpoint URL of AWSIoT is specified.
AccessKeyID=< access key ID >	Required when other than MQTT is specified for Protocol. SecretAccessKey is also required to use this option when MQTT is specified for Protocol. Access key ID of the user who connects it with AWSIoT is specified.
SecretAccessKey=< Secret access key >	Required when other than MQTT is specified for Protocol. AccessKeyID is also required to use this option when MQTT is specified for Protocol. The secret access key of the user who connects it with AWSIoT is specified.
Protocol=< protocol number >	Protocol used to communicate. (1: MQTT, 2: HTTP and 3: MQTT over WebSocket and default: 1)
Thing=<Thing name >	The name of the device connected with AWSIoT is specified. Only CaoController::Execute can be executed when not specifying it.
CA=< root certificate full path >	Root certificate. Only CaoController::Execute is executable when not specifying it.
Certificate=< Thing certificate full path >	Passing of certificate of Thing. Only

[1] 角括弧(“[ ]”)内は省略可能を示します.

	CaoController::Execute is executable when not specifying it.
PrivateKey=< Thing private key full path>	Passing of private key to Thing. Only CaoController::Execute is executable when not specifying it.
QoS=< QoS level>	QoS level. (0: Level 0 and 1: Level 1 and default: 0)

2.2.2. CaoController::Execute method

refer to the 2.3.1, The command name and details that can be used

Format Execute(<bstrCommand:VT_BSTR> [,<vntParam:VARIANT>[,<pVal:VARIANT>]])

<bstrCommand> : In command name
<vntParam> : In parameter
<pVal> : Out acquisition data

2.2.3. CaoController::AddVariable method

The AddVariable method of the CaoController class is a method for making the variable object of each provider. Only the variable of 2.4.1 can be used for the variable identifier.

Format AddVariable(<bstrVariableName:VT_BSTR>[,<bstrOption:VT_BSTR>])

<bstrVariableName> : In variable identifier
<bstrOption> : In optional character string

The following are used for an optional character string.

Table2-3Optional character string of CaoController::AddVariable

Option	Meaning
Topic=< Data kind >	Indispensability. The topic of data is specified.

2.2.4. CaoVariable:get_VariableNames property

The variable of 2.4.1 is acquired.

2.2.5. CaoVariable:get_Value property

Information corresponding to the variable is acquired. About the mounting situation and the acquisition data of each variable, Please refer to 2.4.1.

2.2.6. CaoVariable:put_Value property

Information corresponding to the variable is set. About the mounting situation and the setting data of each variable, Please refer to 2.4.1.

2.2.7. CaoController::OnMessage event

The OnMessage event of the CaoController class is generated according to the seeing and following timing when MQTT or MQTT over WebSocket is used for the communication protocol.

Table2-4Message type

Message type		Generation opportunity
1	AWSIoT data reception	When the data reception is done from AWSIoT, it is generated. (refer to 2.2.7.1)

2.2.7.1. AWSIoT event reception

The data form obtained by the message is shown below.

Number	:	Message type (1)
Value	:	Message received from AWSIoT
DateTime	:	Time stamp
Destination	:	Null
Source	:	Topic
Description	:	Null

2.3. Command list

2.3.1. CaoController class

Table2-5CaoController::Execute command list

Command	Function
RegisterTopic	Received Topic is registered.

UnregisterTopic	The received Topic registration is canceled.	P. 9
CreateThing	Thing is registered in AWSIoT by the specified Thing name. Not available if AccessKeyID / SecretAccekey is not specified when AddController.	P. 10
DeleteThing	The device is deleted from AWSIoT. Not available if AccessKeyID / SecretAccekey is not specified when AddController.	P. 10
GetThingList	Information on the device registered in AWSIoT is acquired. Not available if AccessKeyID / SecretAccekey is not specified when AddController.	P. 10

RegisterTopic

Syntax

object.RegisterTopic(<Data>)

Argument

<Data> = VT_BSTR|Topic received as VT_ARRAY: Subscriber is registered by the array.

Return value

None

Explanation

Topic received as Subscriber is registered. When one or more Topic is registered, OnMessage is generated. It's valid when MQTT or MQTT over WebSocket is used for the communication protocol.

UnregisterTopic

Syntax

object.UnregisterTopic (<Data>)

Argument

<Data> = VT_BSTR|Registration release of Topic received as VT_ARRAY: Subscriber

Return value

None

Explanation

The registration of registered Topic is released. OnMessage is not generated when there is no registered Topic. It's valid when MQTT or MQTT over WebSocket is used for the communication protocol.

CreateThing

Syntax

object.CreateThing (<Data>)

Argument

<Data> = VT_BSTR|VT_ARRAY: Thing name

Array 0: Thing name

Array 1: The preservation of the certificate of made Thing ahead

Return value

VT_BSTR|VT_ARRAY

Array 0: Root certificate passing

Array 1: Thing certificate passing

Array 2: Private key passing

Array 3: The public key passing

Explanation

Thing is registered in AWSIoT by the specified Thing name.

DeleteThing

Syntax

object.DeleteThing(<Data>)

Argument

< Data>= VT_BSTR: Thing name

Return value

None

Explanation

Thing is deleted from AWSIoT.

GetThingList

Syntax

object.GetThingList()

Argument

None

Return value

<Data> = VT_BSTR|Great VT_ARRAY: Thing list

Explanation

Information on Thing registered in AWSIoT is acquired.

2.4. Variable list

2.4.1. CaoController class

Table2-6CaoController class user variable list

Variable identifier	Data type	Explanation	Attribute		Option
			get	put	Topic
*	The option: It applies to the specification of Topic.	The option: It applies to Topic.	-	√	√

Table2-7CaoController class system variable list

Variable identifier	Data type	Explanation	Attribute	
			get	put
@AWSIOT_VERSION	VT_BSTR	Version of AWS IoT C++ Device SDK built into provider.	√	-
@AWSSDK_VERSION	VT_BSTR	.Version of AWSSDK for NET.	√	-
@THINGNAME	VT_BSTR	Thing name.	√	-
@ ENDPOINT	VT_BSTR	Endpoint URL of AWSIoT.	√	-