

Amazon Web Service IoT プロバイダ

Version 1.2.1

ユーザーズ ガイド

January 11, 2023

【備考】

【改版履歴】

バージョン	日付	内容
1.0.0	2017-05-22	初版.
1.1.0	2021-03-19	TLS1.2 対応. EXE 形式に変更.
1.1.1	2021-07-09	OpenSSL を 1.1.1k に変更. Server Name Indication (SNI) 対応.
1.2.0	2021-12-28	Websocket に対応. AddController オプション文字列の変更.
1.2.1	2023-01-11	OpenSSL を 3.0 に変更.

【対応機器】

機種	バージョン	注意事項

目次

1. はじめに	4
2. プロバイダの概要	6
2.1. 概要	6
2.2. メソッド・プロパティ.....	7
2.2.1. CaoWorkspace::AddController メソッド.....	7
2.2.2. CaoController::Execute メソッド.....	8
2.2.3. CaoController::AddVariable メソッド	8
2.2.4. CaoVariable:get_VariableNames プロパティ.....	8
2.2.5. CaoVariable:get_Value プロパティ.....	9
2.2.6. CaoVariable:put_Value プロパティ.....	9
2.2.7. CaoController::OnMessage イベント.....	9
2.2.7.1. AWSIoT イベント受信	9
2.3. コマンド一覧	9
2.3.1. CaoController クラス.....	9
2.4. 変数一覧.....	12
2.4.1. CaoController クラス.....	12

1. はじめに

本書は、Amazon Web Service の AWSIoT に対しデータの送受信を行う CAO プロバイダのユーザーズガイドです。

本書で扱う CAO プロバイダ(CaoProvAWSIoT.exe)を AWSIoT プロバイダと呼びます。

第 2 章に AWSIoT プロバイダの概要、変数の詳細を記載しています。

本プロバイダは、AWSIoT との通信に使用するために AWS IoT C++ Device SDK, AWS SDK for .NET, を使用しています。

これらに関しては、以下のサイトを参照してください。

このプロバイダを使用するためには、「.Net Framework 4.5.2」が必要です。

[AWS IoT C++ Device SDK のサイトリンク]

•AWS IoT C++ Device SDK

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

[AWS IoT C++ Device SDK の著作権とライセンス]

このアプリは Apache License, Version 2.0 のライセンスで配布されている成果物を含んでいます。

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

[AWS SDK for .NET のサイトリンク]

•AWS SDK for .NET

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

[AWS SDK for .NET の著作権とライセンス]

このアプリは Apache License, Version 2.0 のライセンスで配布されている成果物を含んでいます。

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

また、本プロバイダには以下のオープンソースソフトウェアのプログラムが含まれています。

OpenSSL ライセンス

Copyright (C) 1998-2019 The OpenSSL Project.無断での引用、転載を禁じます。

ソースおよびバイナリーの形式での再配布と使用は、変更の有無にかかわらず、以下の条件が満たされた

場合に許可されます。

1. ソースコードを再配布する場合には、上記の著作権表示、この使用条件および以下の免責表示を含める必要があります。
2. バイナリー形式で再配布する場合には、上記の著作権表示、以下の使用条件および免責表示を、配布に際して提供する関連文書および資料に記載する必要があります。
3. このソフトウェアの機能または使用について言及するすべての広告用材料では、次の謝辞を表示する必要があります。「この製品には、OpenSSL ツールキットで使用するために OpenSSL プロジェクトで開発されたソフトウェアが含まれています。(http://www.openssl.org/)」
4. 事前の書面による許可がなければ、「OpenSSL Toolkit」と「OpenSSL Project」の名前を、このソフトウェアから派生した製品の承認または促進に使用してはなりません。書面による許可が必要な場合は、openssl-core@openssl.org にお問い合わせください。
5. OpenSSL Project の事前の書面による許可がなければ、このソフトウェアから派生した製品を「OpenSSL」と呼ぶことはできず、また、それらの製品の名前に「OpenSSL」が含まれていてはなりません。
6. いかなる形の再配布にも、次の謝辞を表示する必要があります。「この製品には、OpenSSL ツールキットで使用するために OpenSSL プロジェクトで開発されたソフトウェアが含まれています。(http://www.openssl.org/)」

OpenSSL Project は、このソフトウェアを特定物として現存するままの状態を提供し、法律上の瑕疵担保責任、商品性の保証および特定目的適合性の保証を含むすべての明示もしくは黙示の保証責任を負いません。起こりうる損害について予見の有無を問わず、「ソフトウェア」を使用したために生じる、直接的、間接的、付随的、特別、懲罰的、または結果的損害（代替の製品またはサービスの調達、データまたは利益の喪失、事業の中断などを含み、他のいかなる場合も含む）については、それが契約、厳格な責任、不法行為（過失の場合もそうでない場合も含む）など、いかなる責任の理論においても、OpenSSL Project およびその寄稿者はその責任を負いません。

この製品には、Eric Young (eay@cryptsoft.com) により作成された暗号化ソフトウェアが含まれています。この製品には、Tim Hudson (tjh@cryptsoft.com) により作成されたソフトウェアが含まれています。

2. プロバイダの概要

2.1. 概要

AWSIoT プロバイダは, AWS IoT C++ Device SDK と AWS SDK for .NET を用いて AWSIoT との通信を行う CAO プロバイダです. そのファイル形式は EXE であり, CAO エンジンから使用時に動的にロードされます. AWSIoT プロバイダを使用するには表 2-1 の方法で登録を行う必要があります. `RegistAsm.bat` および `UnregistAsm.bat` は ORiN2SDK をインストールしたフォルダの下の `DotNet¥BAT` フォルダにあります.

表 2-1 AWSIoT プロバイダ

ファイル名	CaoProvAmazonAWSIoT.exe
ProgID	CaoProv.Amazon.AWSIoT
レジストリ登録	RegistAsm.bat CaoProvAmazonAWSIoT.exe
レジストリ登録の抹消	UnregistAsm.bat CaoProvAmazonAWSIoT.exe

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddController メソッド

AWSIoT プロバイダは AddController 時に通信用の接続パラメータを参照し、AWSIoT との接続を行います。



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

bstrCtrlName : [in] コントローラ名
 bstrProvName : [in] プロバイダ名. 固定値 = "CaoProv.Amazon.AWSIoT"
 bstrPcName : [in] プロバイダの実行マシン名
 bstrOption : [in] オプション文字列

表 2-2 CaoWorkspace::AddController のオプション文字列

オプション ¹	説明
Endpoint=<AWSIoT EndPoint>	必須. AWSIoT の Endpoint URL を指定します.
AccessKeyID=<アクセスキーID>	Protocol に MQTT 以外を指定した際に必須. Protocol に MQTT を指定した際にこのオプションを使用するには SecretAccessKey も必須. AWSIoT へ接続するユーザのアクセスキーID を指定します.
SecretAccessKey =<シークレットアクセスキー>	Protocol に MQTT 以外を指定した際に必須. Protocol に MQTT を指定した際にこのオプションを使用するには AccessKeyID も必須. AWSIoT へ接続するユーザのシークレットアクセスキーを指定します.
Protocol=[<プロトコル番号>]	通信に用いるプロトコル. (1:MQTT, 2:HTTP, 3: MQTT over WebSocket, デフォルト:1)
Thing=[<Thing 名>]	AWSIoT へ接続するデバイスの名前を指定します. 指定しない場合は CaoController::Execute のみ実行できます.
CA=[<ルート証明書フルパス>]	ルート証明書. 指定しない場合は CaoController::Execute のみ実行可能です.
Certificate=[<Thing 証明書フルパス>]	Thing の証明書のパス. 指定しない場合は

¹ 角括弧("[]")内は省略可能を示します.

	CaoController::Execute のみ実行可能です。
PrivateKey=[<Thing プライベートキーフルパス>]	Thing のプライベートキーのパス. 指定しない場合は CaoController::Execute のみ実行可能です。
QoS=[<QoS レベル>]	QoS レベル. (0:レベル 0, 1:レベル 1, デフォルト:0)

2.2.2. CaoController::Execute メソッド

使用できるコマンド名と詳細は 2.3.1 を参考にしてください。

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

<bstrCommand> : [in] コマンド名
 <vntParam> : [in] パラメータ
 <pVal> : [out] 取得データ

2.2.3. CaoController::AddVariable メソッド

CaoController クラスの AddVariable メソッドは、それぞれのプロバイダの変数オブジェクトを作成するためのメソッドです。変数名には、2.4.1 の変数のみ使用することができます。

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

<bstrVariableName> : [in] 変数名
 <bstrOption> : [in] オプション文字列

オプション文字列には以下のものを用います。

表 2-3 CaoController::AddVariable のオプション文字列

オプション	意味
Topic=<データ種類>	必須. データのトピックを指定.

2.2.4. CaoVariable:get_VariableNames プロパティ

2.4.1 の変数を取得します。

2.2.5. CaoVariable:get_Value プロパティ

変数に対応する情報を取得します。各変数の実装状況および取得データについては、2.4.1を参照して下さい。

2.2.6. CaoVariable:put_Value プロパティ

変数に対応する情報を設定します。各変数の実装状況および設定データについては、2.4.1を参照して下さい。

2.2.7. CaoController::OnMessage イベント

通信プロトコルに MQTT, MQTT over WebSocket を使用した時のみ、以下のタイミングで CaoController クラスの OnMessage イベントが発生します。

表 2-4 メッセージ種別

メッセージ種別		発生契機
1	AWSIoT データ受信	AWSIoT からデータ受信が行われた際に発生します。 (参照:2.2.7.1)

2.2.7.1. AWSIoT イベント受信

メッセージで得られるデータ形式を以下に示します。

Number	:	メッセージ種別 (1)
Value	:	AWSIoT から受信したメッセージ
DateTime	:	タイムスタンプ
Destination	:	Null
Source	:	トピック
Description	:	Null

2.3. コマンド一覧

2.3.1. CaoController クラス

表 2-5 CaoController::Execute コマンド一覧

コマンド	機能	
RegisterTopic	受信する Topic を登録する.	P. 10
UnregisterTopic	受信する Topic 登録を取り消す.	P. 10
CreateThing	指定した Thing 名で AWSIoT に Thing を登録する. AddController 時に	P. 10

DeleteThing	AccessKeyID/SecretAccekey を指定しない場合は使用不可。 AWSIoT からデバイスを削除する。AddController 時に P.11 AccessKeyID/SecretAccekey を指定しない場合は使用不可。
GetThingList	AWSIoT に登録されたデバイスの情報を取得する。AddController 時に P.11 AccessKeyID/SecretAccekey を指定しない場合は使用不可。

RegisterTopic

構文	<code>object. RegisterTopic (<Data>)</code>
引数	<Data> = VT_BSTR VT_ARRAY: Subscriber として受け取る Topic を配列で登録する。
戻り値	なし
説明	Subscriber として受け取る Topic を登録する。1 つ以上の Topic が登録された場合に OnMessage が発生する。通信プロトコルに MQTT, MQTT over WebSocket を使用した時のみ有効。

UnregisterTopic

構文	<code>object. UnregisterTopic (<Data>)</code>
引数	<Data> = VT_BSTR VT_ARRAY: Subscriber として受け取る Topic の登録解除
戻り値	なし
説明	登録した Topic の登録を解除する。登録した Topic がない場合は OnMessage は発生しない。通信プロトコルに MQTT, MQTT over WebSocket を使用した時のみ有効。

CreateThing

構文	<code>object. CreateThing (<Data>)</code>
引数	<Data> = VT_BSTR VT_ARRAY: Thing 名 Array[0]: Thing 名 Array[1]: 作成した Thing の証明書の保存先

戻り値	VT_BSTR VT_ARRAY Array[0]: ルート証明書パス Array[1]: Thing 証明書パス Array[2]: プライベートキーパス Array[3]: パブリックキーパス
説明	指定した Thing 名で AWSIoT に Thing を登録する.

DeleteThing

構文	<code>object.DeleteThing(<Data>)</code>
引数	<Data> = VT_BSTR: Thing 名
戻り値	なし
説明	AWSIoT から Thing を削除する.

GetThingList

構文	<code>object.GetThingList()</code>
引数	なし
戻り値	<Data> = VT_BSTR VT_ARRAY: Thing 名一覧
説明	AWSIoT に登録された Thing の情報を取得する.

2.4. 変数一覧

2.4.1. CaoController クラス

表 2-6 CaoController クラス ユーザ変数一覧

変数名	データ型	説明	属性		オプション
			get	put	Topic
*	オプション:Topic の指定に準ず る.	オプション:Topic に準ずる.	-	○	○

表 2-7 CaoController クラス システム変数一覧

変数名	データ型	説明	属性	
			get	put
@AWSIOT_VERSION	VT_BSTR	プロバイダに組み込まれた AWS IoT C++ Device SDK のバージョン.	○	-
@AWSSDK_VERSION	VT_BSTR	.NET 用 AWSSDK のバージョン.	○	-
@THINGNAME	VT_BSTR	Thing 名.	○	-
@ ENDPOINT	VT_BSTR	AWSIoT の Endpoint URL.	○	-