

# Microsoft Azure IoT プロバイダ

Version 1.5.1

## ユーザーズ ガイド

Oct 22, 2024

【備考】

## 【改版履歴】

バージョン	日付	内容
1.0.0	2016-09-09	初版.
1.0.1	2016-10-04	CaoWorkspace::AddController のオプションを変更. CaoController::Execute メソッドを追加. CaoController クラス システム変数の追加.
1.0.2	2017-07-31	CaoController::Execute コマンド一覧の誤記修正.
1.0.3	2019-10-28	Azure SDK 置き換えに伴うライセンス情報の追記.
1.1.0	2020-01-29	Azure IoT Edge に対応. ダイレクトメソッドに対応. デバイスツインに対応. Azure IoT Hub Device Provisioning Service に対応.
1.2.0	2020-02-03	AzureIoT Central 対応
1.3.0	2021-03-19	TLS1.2 に対応. ファイル形式を exe に変更. Websocket に対応.
	2021-09-08	ユーザ変数 DataType 一覧の誤記修正.
	2021-11-22	CaoController::Execute 説明文の追記.
1.3.1	2022-02-15	不具合修正.
1.4.0	2022-04-28	証明書を使用してのデバイス接続に対応. DPS 接続時の Endpoint 指定に対応.
1.5.0	2023-03-10	DPS 接続時の証明書認証に対応.
1.5.1	2024-10-22	API バージョン取得箇所の不具合修正. 全角文字列送信時に文字化けする問題を修正.

## 【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
2.2.1.1. ConnStr オプション.....	9
2.2.2. CaoController::Execute メソッド.....	10
2.2.3. CaoController::AddVariable メソッド.....	10
2.2.4. CaoVariable:get_VariableNames プロパティ.....	10
2.2.5. CaoVariable:get_Value プロパティ.....	10
2.2.6. CaoVariable:put_Value プロパティ.....	11
2.2.7. CaoController::OnMessage イベント.....	11
2.2.7.1. IoT Hub イベント受信.....	11
2.3. コマンド一覧.....	12
2.3.1. CaoController クラス.....	12
2.4. 変数一覧.....	14
2.4.1. CaoController クラス.....	14

## 1. はじめに

本書は、Microsoft Azure IoT Hub 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(CaoProvAzureIoT.exe)を AzureIoT プロバイダと呼びます。

第 2 章に AzureIoT プロバイダの概要、変数の詳細を記載しています。

AzureIoT プロバイダでは、通信に使用するために Azure IoT SDK 等の OSS を使用しています。

Azure IoT SDK に関しては、以下のサイトを参照してください。

[関連サイトリンク]

<https://github.com/Azure/azure-iot-sdks>

[Azure IoT SDK の著作権とライセンス]

Copyright (c) Microsoft Corporation

All rights reserved.

MIT License

<https://github.com/Azure/azure-iot-sdk-csharp/blob/master/LICENSE>

[DotNetty の著作権とライセンス]

Copyright (c) Microsoft Corporation

All rights reserved.

MIT License

<https://github.com/Azure/DotNetty/blob/dev/LICENSE.txt>

[Npgsql の著作権とライセンス]

Copyright (c) 2003 Carlos Guzmán Álvarez

All rights reserved.

MIT License

<https://github.com/danzel/Npgsql/blob/master/lib/Mono.Security/2.0/Mono.Security.Protocol.Tls.license>

[Newtonsoft.Json の著作権とライセンス]

Copyright (c) 2007 James Newton-King

All rights reserved.

MIT License

<https://raw.githubusercontent.com/JamesNK/Newtonsoft.Json/master/LICENSE.md>

[OData .NET Libraries の著作権とライセンス]

Copyright (c) 2018 Microsoft. All rights reserved.

All rights reserved.

MIT License

<https://raw.githubusercontent.com/OData/odata.net/master/LICENSE.txt>

[Microsoft Azure Storage 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/Azure/azure-storage-net/blob/master/LICENSE.txt>

[PCLCrypto の著作権とライセンス]

このアプリは Microsoft Public License (Ms-PL) 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://raw.githubusercontent.com/AArnott/PCLCrypto/313d8a787a/LICENSE>

[Validation の著作権とライセンス]

このアプリは Microsoft Public License (Ms-PL) 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://raw.githubusercontent.com/AArnott/Validation/69e6a2c4f3/LICENSE.txt>

[Microsoft.Extensions 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/aspnet/Extensions/blob/master/LICENSE.txt>

[PInvoke.net の著作権とライセンス]

Copyright (c) 2015 Andrew Arnott

All rights reserved.

MIT License

<https://github.com/AArnott/pinvoke/blob/master/LICENSE>

## 2. プロバイダの概要

### 2.1. 概要

AzureIoT プロバイダは、Azure IoT SDK を用いて IoT Hub との双方向通信を行う 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。AzureIoT 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 AzureIoT プロバイダ

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

## 2.2. メソッド・プロパティ

### 2.2.1. CaoWorkspace::AddController メソッド

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

**書式** 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.Microsoft.AzureIoT"  
 bstrPcName : [in] プロバイダの実行マシン名  
 bstrOption : [in] オプション文字列

以下にオプション文字列に指定するリストを示します。

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

オプション <sup>1</sup>	説明
ConnMode=[<DPS 使用/未使用>]	IoT Hub との接続時に Device Provisioning Service を使用するかを指定する。 (0:使用しない, 1:使用する, デフォルト:0)
ConnStr=(<IoT Hub Connection String   Device Connection String>)	DPSMode 未使用時に必須. IoT Hub 接続文字列または デバイスの接続文字列. ()で囲む. デバイスの接続文字列であった場合には DeviceId オプションは無視する. (参照:2.2.1.1) 指定しない場合は証明書を使用してのデバイス接続になる.
DeviceId=[<DeviceID>]	DPSMode 使用時もしくは証明書を使用してのデバイス接続時に必須. クライアントのデバイス ID. ConnStr に IoT Hub 接続文字列を指定し, DeviceID が空であった場合には, CaoController::Execute("DeviceCreate")でデバイス登録するまで IoT Hub とのメッセージ・BLOB 通信不可.
HostName=[<HostName>]	Device Provisioning Service 未使用時に, 証明書

<sup>1</sup> 角括弧("[ ]")内は省略可能を示します。

	でのデバイス接続時に必須.
Certificate=[<デバイス証明書のフルパス>]	証明書(.pfx)を使用してのデバイス接続時に必須.
Passphrase=[< Passphrase >]	証明書使用時のパスフレーズ. 指定しなければ証明書をパスフレーズなしで使用する.
IdScope=[<IDScope>]	DPSMode 使用時に必須. デバイスの ID スコープを指定する.
SAS=[(<Shared Access Signature>)]	DPSMode 使用時にデバイスの Shared Access Signature を指定する.
DPSEndpoint=[<Endpoint>]	DPSMode 使用時に指定. 指定しない場合はデフォルト値を使用する. (デフォルト: global.azure-devices-provisioning.net)
Protocol=[<プロトコル番号>]	(1:MQTT, 2:HTTP, 3:AMQP, 4:MQTT over WebSocket, 5: AMQP over WebSocket, デフォルト:1)
Method01=[<ダイレクトメソッド名>] (同様に Method02 - Method99 まで指定可能. 連番が途切れたオプション文字列までのダイレクトメソッド名を登録)	ダイレクトメソッド使用時に, 実装するダイレクトメソッド名を指定する. 指定したダイレクトメソッドを受け取った際には正常動作を返す.

### 2.2.1.1. ConnStr オプション

以下に ConnStr オプションのパラメータ文字列を示します。

“ConnStr=<IoT Hub Connection String>”

< IoT Hub Connection String : IoT Hub 接続文字列 もしくは デバイス接続文字列.  
String >

IoT Hub 接続文字列とデバイス接続文字列は Azure Portal から確認することができます。



また、デバイス接続文字列は CaoController のシステム変数でも確認することができます。

### 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。コマンドを使用するには AddController 時の ConnStr オプションにて、IoT Hub 接続文字列を指定している必要があります。

**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が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て、データ送信の同期設定を指定することができます。

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

オプション	意味
DataType[=<変数種類>]	変数の種類を指定。 (MSG : メッセージ , BLOB : BLOB , Properties : デバイスツイン Properties , Desired : デバイスツイン Desired, Reported : デバイスツイン Reported, デフォルト:MSG)

### 2.2.4. CaoVariable:get\_VariableNames プロパティ

2.3.1 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、2.3.1 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

以下の契機で 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	IoT Hub データ受信	IoT Hub 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. IoT Hub 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:デバイスへのメッセージ, 2:ダイレクトメソッド, 3:デバイスツイン)
Value	:	IoT Hub から受信したメッセージ
DateTime	:	タイムスタンプ
Destination	:	Null
Source	:	Null
Description	:	デバイスへのメッセージ受信時:Null, ダイレクトメソッド受信時:ダイレクトメソッドコマンド名

## 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	
CreateDevice	指定したデバイス ID で IoT Hub にデバイスを登録する。	P. 12
EditDevice	IoT Hub 上のデバイス登録情報を更新する。	P. 12
DeleteDevice	IoT Hub からデバイスを削除する。	P. 13
GetDeviceList	IoT Hub に登録されたデバイスの情報を取得する。	P. 13

## CreateDevice

### 構文

`object.CreateDevice (<Data>)`

### 引数

<Data> = VT\_BSTR または VT\_EMPTY: デバイス ID(省略可能)

### 戻り値

なし

### 説明

指定したデバイス ID で IoT Hub にデバイスを登録する。デバイス ID を指定しない場合にはデバイス ID が自動生成される。  
(通信プロトコルとして HTTP 選択時のみ使用可能です)

## EditDevice

### 構文

`object.EditDevice (<Data>)`

### 引数

<Data> = VT\_BSTR|VT\_ARRAY: デバイス情報

Array[0]: デバイス ID

Array[1]: プライマリキー

Array[2]: セカンダリキー

### 戻り値

なし

### 説明

IoT Hub 上のデバイス登録情報を更新する。  
(通信プロトコルとして HTTP 選択時のみ使用可能です)

---

## DeleteDevice

---

構文	<code>object.DeleteDevice(&lt;Data&gt;)</code>
引数	<Data> = VT_BSTR: デバイス ID
戻り値	なし
説明	IoT Hub からデバイスを削除する。 (通信プロトコルとして HTTP 選択時のみ使用可能です)

---

## GetDeviceList

---

構文	<code>object.GetDeviceList()</code>
引数	なし
戻り値	<Data> = VT_ARRAY VT_ARRAY: デバイス情報一覧 Array[*][0]: デバイス ID Array[*][1]: プライマリキー Array[*][2]: セカンダリキー Array[*][3]: X.509 プライマリサムプリント Array[*][4]: X.509 セカンダリサムプリント Array[*][5]: 接続文字列 Array[*][6]: 接続状態 Array[*][7]: 最終アクティビティ時間 Array[*][8]: 最終接続状態更新時間 Array[*][9]: 最終状態更新時間 Array[*][10]: メッセージ数 Array[*][11]: 状態 Array[*][12]: 停止理由
説明	IoT Hub に登録されたデバイスの情報を取得する。

## 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	DataType
*	オプション :	オプション:DataType に準ずる. DataType の指定 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
@SDK_VERSION	VT_BSTR	プロバイダ開発に使用した SDK のバージョン.	○	-
@API_VERSION	VT_BSTR	IoT SDK で使用している API のバージョン.	○	-
@DEVICEID	VT_BSTR	デバイス ID.	○	-
@PRIMARY_KEY	VT_BSTR	デバイスのプライマリキー.	○	-
@SECONDARY_KEY	VT_BSTR	デバイスのセカンダリキー.	○	-
@DEVICE_CONSTRING	VT_BSTR	デバイスの接続文字列.	○	-

表 2-8 ユーザ変数 DataType 一覧

オプション名	データ型	説明	属性	
			get	put
MSG	VT_UI1 VT_AR RAY または VT_BSTR	IoT Hub にメッセージデータを同期モードで送信 する.	-	○

BLOB	VT_BSTR VT_ARRAY	<p>IoT Hub に BLOB データを同期モードで送信する。</p> <p>ファイル情報を以下の順番で配列に格納します。</p> <p>&lt;送信元ファイルパス&gt; [&lt;送信先ファイル名&gt;]</p> <p>※送信先ファイル名は省略可能です。送信先ファイル名を省略した場合には、送信元ファイルのファイル名で保存されます。</p> <p>※送信先ファイル名には”(フォルダ名)¥(ファイル名)”の様にフォルダを指定することができます。</p> <p>※送信先ファイル名で”(フォルダ名)¥”とファイル名無しでオプションを渡した場合は、クラウド上に”(フォルダ名)¥(送信元ファイル名)”で保存されます。</p> <p>※クラウド上に同名ファイルがある場合は上書きされます。</p>	-	○
Properties	VT_BSTR	IoT Hub の Device Twin Properties を読み取ります。一部 null で返る値があります。	○	-
Desired	VT_BSTR	IoT Hub の Device Twin Desired 値を読み取ります。	○	-
Reported	VT_BSTR	IoT Hub の Device Twin Reported 値を読み書きします。	○	○