

Microsoft Azure IoT provider

Version 1.5.1

User's Guide

Oct 22, 2024

[Notes]

[Revision History]

Version	Date	Description
1.0.0	2016-09-09	First edition
1.0.1	2016-10-04	Change CaoWorkspace::AddController options. Added CaoController::Execute method. Adding CaoController Class System Variables.
1.0.2	2017-07-31	Error in the list of CaoController::Execute commands was corrected.
1.0.3	2019-10-28	Addition of licensing for Azure SDK replacement.
1.1.0	2020-01-29	Supports Azure IoT Edge. Supports direct methods. Supports device twin. Supports Azure IoT Hub Device Provisioning Service.
1.2.0	2020-02-03	Supports AzureIoT Central.
1.3.0	2020-03-19	Supports TLS 1.2. Change to EXE format. Supports Websocket.
	2021-09-08	Fixed list of user variables DataType.
	2021-11-22	Added CaoController::Execute description.
1.3.1	2022-02-15	Bug fix.
1.4.0	2022-04-28	Supports device connection using certificate. Supports Endpoint designation at DPS connection.
1.5.0	2023-03-10	Supports certificate authentication for DPS connections.
1.5.1	2024-10-22	Fixed an issue with retrieving the API version. Fixed an issue where full-width characters would become garbled when sent.

[Supported Devices]

Model	Version	POINTS OF CAUTION

Contents

1. Introduction	4
2. Provider Overview	6
2.1. Introduction	6
2.2. Method Properties	7
2.2.1. CaoWorkspace::AddController method	7
2.2.1.1. ConnStr Options	9
2.2.2. CaoController::Execute method	10
2.2.3. CaoController::AddVariable method	10
2.2.4. CaoVariable: get_VariableNames Property	10
2.2.5. CaoVariable: get_Value Property	10
2.2.6. CaoVariable: put_Value	11
2.2.7. CaoController::OnMessage events	11
2.2.7.1. Receiving IoT Hub events	11
2.3. Command list	12
2.3.1. CaoController classes	12
2.4. List of Variables	14
2.4.1. CaoController classes	14

1. Introduction

This document is the user's guide of the CAO provider that sends and receives data to and from Microsoft Azure IoT Hub.

The CAO provider (CaoProvAzureIoT.exe) described in this document is called AzureIoT provider.

Section 2 provides an overview of AzureIoT providers and more information about variables.²

Azure IoT providers use OSSs such as Azure IoT SDK to communicate with each other.

For more Azure IoT SDK, see:

Related Site Links

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

Azure IoT SDK Copyrights and Licenses

Copyright (c) Microsoft Corporation

All rights reserved.

MIT License

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

DotNetty Copyrights and Licenses

Copyright (c) Microsoft Corporation

All rights reserved.

MIT License

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

Npgsql Copyrights and Licenses

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 Copyrights and Licenses

Copyright (c) 2007 James Newton-King

All rights reserved.

MIT License

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

OData.NET Libraries Copyrights and Licenses

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 Copyrights and Licenses

This app contains artifacts distributed with Apache License, Version 2.0 licenses.

<https://github.com/Azure/azure-storage-net/blob/master/LICENSE.txt>

PCLCrypto Copyrights and Licenses

This app contains artifacts distributed with Microsoft Public License (Ms-PL) licenses.

<https://raw.githubusercontent.com/AArnott/PCLCrypto/313d8a787a/LICENSE>

Validation Copyrights and Licenses

This app contains artifacts distributed with Microsoft Public License (Ms-PL) licenses.

<https://raw.githubusercontent.com/AArnott/Validation/69e6a2c4f3/LICENSE.txt>

Microsoft.Extensions Copyrights and Licenses

This app contains artifacts distributed with Apache License, Version 2.0 licenses.

<https://github.com/aspnet/Extensions/blob/master/LICENSE.txt>

PInvoke.net Copyrights and Licenses

Copyright (c) 2015 Andrew Arnott

All rights reserved.

MIT License

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

2. Provider Overview

2.1. Introduction

A Azure IoT provider is a CAO provider that uses Azure IoT SDK to communicate bidirectionally with a IoT Hub. Its file format is EXE, which is loaded dynamically from the CAO engine when used. To use AzureIoT providers, you must register as shown in Table 2-1. RegistAsm. bat and UnregistAsm. bat are located in DotNet ¥BAT folder under the folder where you installed ORiN2SDK. Tables 2-1

Tables 2-1 AzureIoT Providers

File name	CaoProv.Microsoft.AzureIoT.exe
ProgID	CaoProv.Microsoft.AzureIoT
Registry registration	RegistAsm.bat CaoProv.Microsoft.AzureIoT.exe
Deletion of Registry Registration	UnregistAsm.bat CaoProv.Microsoft.AzureIoT.exe

2.2. Method Properties

2.2.1. CaoWorkspace::AddController method

Azure IoT providers connect to Azure IoT Hub by referring to the connection parameters for communication at the time of AddController.

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

- BstrCtrlName : [in] Controller name
- BstrProvName : [in] Provider name. Fixed value = "CaoProv.Microsoft.AzureIoT"
- BstrPcName : [in] The name of the provider's running machine
- BstrOption : [in] option string

The following is the list specified in the option string.

Optional strings for table 2-2 CaoWorkspace::AddController

OPTIONS ¹	Description
ConnMode = [<DPS used/not used>]	Specifies Device Provisioning Service to use when connecting to IoT Hub. (0: Not used, 1: Enable, Default: 0)
ConnStr=(<IoT Hub Connection String Device Connection String>)	Required when DPSPMode is not used. IoT Hub connection string or device connection string. Enclose in parentheses. If it is a device connect string, DeviceId option is ignored.(Refer to 2.2.1.1) If not specified, it will be device connection using certificate.
DeviceId=[<DeviceID >]	Required when using DPSPMode or device connection using certificate. If IoT Hub connection string is specified in the device ID. ConnStr of the client and DeviceID is empty, message BLOB communication with IoT Hub is not possible until the device is registered with "CaoController::Execute("DeviceCreate").
HostName=[<HostName>]	Required when Device Provisioning Service is not used and when connecting a device with a

¹ Square brackets ("[]") indicate optional characters.

	certificate.
Certificate=[<Device certificate full path>]	Required when device connection using certificate (. pfx).
Passphrase=[< Passphrase >]	Passphrase for the certificate. If not specified, the certificate is used without a passphrase.
IdScope=[<IdScope >]	Required when using DPSMode. Specifies the ID scope of the device.
SAS=[(<Shared Access Signature>)]	Required when using DPSMode. Specifies Shared Access Signature of devices.
DPSEndpoint=[<Endpoint>]	It can be used when using DPSMode. If not specified, the default value will be used.(Default: global.azure-devices-provisioning.net)
Protocol = [<protocol-number>]	The protocol used for communication. (1:MQTT, 2:HTTP, 3:AMQP, 4:MQTT over Websocket, 5: AMQP over Websocket, Default: 1)
Method01=[<direct-method-name>] (Similarly, up to Method02-Method99 can be specified. Register the name of the direct method up to the option string where the serial number was interrupted.)	Specifies the name of the direct method to be implemented when the direct method is used. When the specified direct method is received, normal operation is returned.

2.2.1.1. ConnStr Options

The following are the parameter strings for ConnStr options:

"ConnStr=<IoT Hub Connection String>"

< IoT Hub Connection String > : IoT Hub connection string or device connection string.
String >

You can check IoT Hub connection string and the device connection string from Azure Portal.

Shared access keys

Primary key ⓘ

b//1XcuC4xjTZ8Qr0TcdAjdHsZnHhtaI3E3C

Secondary key ⓘ

HHj+HHI1wibmUgNRqcjncsn7b9OpOaA;

Connection string—primary key ⓘ

HostName=demo-iothub.azure-devices.n

Connection string—secondary key ⓘ

HostName=demo-iothub.azure-devices.n

You can also see the device-connection strings in CaoController system variables.

2.2.2. CaoController::Execute method

See section 2.3.1 for the command names and details.2.3.1. To use the command, it is necessary to specify the IoT Hub connection string in the ConnStr option at the time of AddController.

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

- <bstrCommand> : [in] Command name
- <vntParam> : [in] Parameters
- <pVal> : [out] Acquired data

2.2.3. CaoController::AddVariable method

AddVariable method of CaoController class is a method for each provider to create variables objects of their own. Only the 2.4.1 variable can be used in the variable name.2.4.1

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

- <bstrVariableName> : [in] Variable name
- <bstrOption> : [in] option string

The following options can be used to specify the synchronization settings for data transmission:

Optional strings for table 2-3 CaoController::AddVariable

OPTIONS	Meaning
DataType[=<variable type>]	Specifies the type of variable. (MSG:Message, BLOB:BLOB,Properties:Device Twin Properties,Desired:Device Twin Desired,Reported:Device Twin Reported, Default:MSG)

2.2.4. CaoVariable: get_VariableNames Property

Get the variables of 2.3.1

2.2.5. CaoVariable: get_Value Property

Get information corresponding to a variable. Refer to 2.3.1 for the implementation status of each variable and

the acquired data.2.3.1

2.2.6. CaoVariable: put_Value

Sets the information corresponding to the variable. Refer to 2.4.1 for the implementation status of each variable and the setting data.2.4.1

2.2.7. CaoController::OnMessage events

OnMessage events of CaoController classes occur at the following events.

Table 2-4 Message types

Message type		Trigger of occurrence
1	Receive IoT Hub	Occurs when data is received from a IoT Hub. (Refer to 2.2.7.1)2.2.7.1

2.2.7.1. Receiving IoT Hub events

The following table shows the data format of the message.

- Number : Message type (1: message to device, 2: direct method, 3: device twin)
- Value : Messages received from IoT Hub
- DateTime : Time stamp
- Destination : Null
- Source : Null
- Description : When receiving a message to the device: Null, when receiving a direct method: Name of a direct method command

2.3. Command list

2.3.1. CaoController classes

List of 2-5 CaoController::Execute Commands

Command	Function	
CreateDevice	Registers a device to IoT Hub with the specified device ID.	P.1112
EditDevice	Updates the registered devices on IoT Hub.	P.1112
DeleteDevice	Remove devices from IoT Hub.	P.1212
GetDeviceList	Gets the info about the devices registered in IoT Hub.	P.1213

CreateDevice

Syntax *Object.CreateDevice* (<Data>)

Argument <Data>= VT_BSTR or VT_EMPTY: Device identifier (optional)

Return None

Values

Description Registers a device to IoT Hub with the specified device ID. When the device ID is not specified, the device ID is automatically generated.
(Available only when HTTP is selected as the communication protocol.)

EditDevice

Syntax *Object.EditDevice* (<Data>)

Argument <Data> = VT_BSTR|VT_ARRAY: Device-Information

Array[0]: Device ID
Array[1]: primary key
Array[2]: secondary key

Return None

Values

Description Updates the registered devices on IoT Hub.
(Available only when HTTP is selected as the communication protocol.)

DeleteDevice

Syntax *Object.DeleteDevice*(<Data>)

Argument <Data> = VT_BSTR: Device-ID

Return	None
Values	
Description	Remove devices from IoT Hub. (Available only when HTTP is selected as the communication protocol.)

GetDeviceList

Syntax	<i>Object.GetDeviceList()</i>
Argument	None
Return	<Data>= VT_ARRAY VT_ARRAY: List of device information
Values	Array[*][0]: Device ID Array[*][1]: primary key Array[*][2]: secondary key Array[*][3]: X.509 primary thumbprint Array[*][4]: Secondary X.509 thumbprint Array[*][5]: connection string Array[*][6]: connection status Array[*][7]: Last Activity Time Array[*][8]: Last connection status update time Array[*][9]: last status update time Array[*][10]: number of messages Array[*][11]: Status Array[*][12]: reason for termination
Description	Gets the info about the devices registered in IoT Hub.

2.4. List of Variables

2.4.1. CaoController classes

Table List of 2-6 CaoController Class User Variables

Variable name	Data Type	Description	Attribute		OPTIONS
			Get	Put	DataType
*	Optional: Follow DataType specification.	Optional: According to DataType.	-	○	○

Table List of 2-7 CaoController Class System Variables

Variable name	Data Type	Description	Attribute	
			Get	Put
@SDK_VERSION	VT_BSTR	The version of the SDK used to develop the provider.	○	-
@API_VERSION	VT_BSTR	Versions of the APIs used in IoT SDK.	○	-
@DEVICEID	VT_BSTR	Device ID.	○	-
@PRIMARY_KEY	VT_BSTR	The primary key for the device.	○	-
@SECONDARY_KEY	VT_BSTR	Secondary key for the device.	○	-
@DEVICE_CONSTRING	VT_BSTR	The connection string for the device.	○	-

Table 2-8 DataType List for User Variables

Option name	Data Type	Description	Attribute	
			Get	Put
MSG	VT_UI1 VT_AR RAY Or VT_BSTR	Send the message data to IoT Hub in synchronous mode.	-	○

BLOB	VT_BSTR VT_ARRAY	<p>Send BLOB data to IoT Hub in synchronous mode.</p> <p>The file information is stored in an array in the following order:</p> <p><source file path> [<destination file name>]</p> <p>※The destination file name is optional. When the destination file name is omitted, the file is saved with the file name of the source file.</p> <p>※ The destination file name can be specified as "(folder name)¥(file name)".</p> <p>※If you pass "(folder name)¥" as the destination file name and no file name as the option, the file is saved in the cloud as "(folder name)¥(source file name)".</p> <p>※If a file with the same name exists on the cloud, it will be overwritten.</p>	-	○
Properties	VT_BSTR	Read Device Twin Properties of IoT Hub. Some values are returned as null.	○	-
Desired	VT_BSTR	Read Device Twin Desired of IoT Hub.	○	-
Reported	VT_BSTR	Read and write Device Twin Reported of IoT Hub.	○	○