

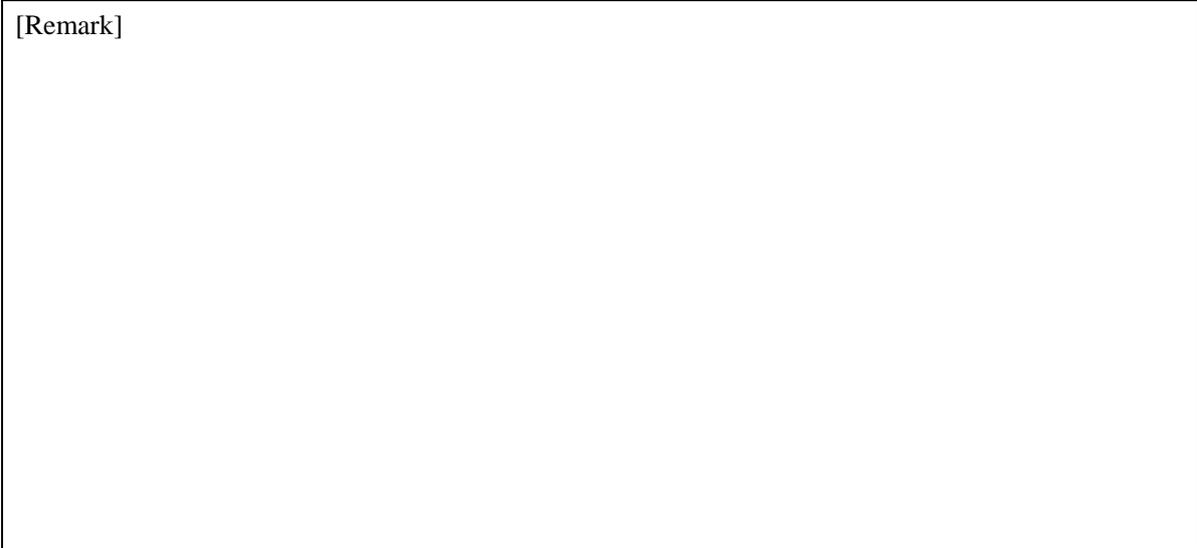
Microsoft Azure Storage providers

Version 1.1.3

User's Guide

November 4, 2022

[Remark]



[Revision History]

Version	Date	Description
1.0.0	2017-07-26	First edition
1.0.1	2020-06-12	Replace SDK.
1.1.0	2021-06-02	Supports TLS 1.2. Change to EXE format.
	2021-07-08	Adjusting the appearance of the user's guide.
1.1.1	2021-08-25	"CaoVariable :: put_Value" bug fix.
1.1.2	2021-12-13	Bug Fix. Document Modification.
1.1.3	2022-11-04	Bub Fix.

[Compatible devices]

Model	Version	Notes

Contents

1. Introduction	4
2. Provider Overview	5
2.1. Introduction	5
2.2. Method properties	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. CaoController::AddFile method.....	8
2.2.5. CaoController::get_FileNames Property	10
2.2.6. CaoFile::AddFile method	10
2.2.7. CaoFile::Copy method	10
2.2.8. CaoFile::Delete method.....	12
2.2.9. CaoFile::Move method.....	12
2.2.10. CaoFile::get_FileNames Property	12
2.2.11. CaoFile::get_Attribute Property	12
2.2.12. CaoFile::get_Path property.....	13
2.2.13. CaoFile::get_Size	13
2.2.14. CaoFile::get_Type Property.....	13
2.2.15. CaoFile::get_DateLastModified Property	13
2.2.16. CaoFile::get_Value Property	13
2.2.17. CaoFile::put_Value Property	13
2.2.18. CaoVariable::get_VariableNames Property	13
2.2.19. CaoVariable::get_Value Property	13
2.2.20. CaoVariable::put_Value Property	14
2.3. Command list	14
2.3.1. CaoController classes	14
2.4. Variable list.....	18
2.4.1. CaoController classes	18

1. Introduction

This document is the user's guide for CAO providers who send and receive data to and from Storage in Microsoft Azure.

The CAO-providers (CaoProvAzureStorage.exe) covered in this document are called AzureStorage providers. Section 2 provides an overview of AzureStorage providers and detailed information about the variables.2

This provider uses Microsoft Azure Storage SDK for.NET to communicate with AzureStorage.

For more information about these, see the following site:

To use this provider, "Net Framework 4.5.2" is required.

[Microsoft Azure Storage SDK for.NET site link]

- Microsoft Azure Storage SDK for .NET

<https://github.com/Azure/azure-storage-net>

[Microsoft Azure Storage SDK for.NET Copyright and Licensing]

This app contains artifacts that are licensed for Apache License, Version 2.0.

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

2. Provider Overview

2.1. Introduction

Azure Storage providers are CAO providers that communicate with Microsoft Azure Storage SDK for.NET Blob Storage. The file format is a EXE that is dynamically loaded when used by the CAO engine. To use AzureStorage provider, you must register in the manner of Table 2-1. RegistAsm.bat and UnregistAsm.bat are in DotNet¥BAT folders below ORiN2SDK installed.

Table 2-1 AzureStorage Providers

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

2.2. Method properties

2.2.1. CaoWorkspace::AddController method

AzureStorage providers connect to AzureStorage by referring to the connection parameters for communication during AddController.

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

- BstrCtrlName : [in] Controller name
- BstrProvName : [in] Provider name. Fixed value = "CaoProv.Microsoft.AzureStorage"
- BstrPcName : [in] Provider's running machine name
- BstrOption : [in] Option string

Table 2-2 CaoWorkspace::AddController Option-String

Option	Description
RootName=<BlobContainer name/ Share name>	Mandatory. Specify the following as specified in Service: For BlobStorage: Specifies BlobContainer. For FileStorage: Specifies Share.
StorageAccount = <storage account name>	Mandatory. Specify Storage account-name.
AccountKey=<Storage Account Key>	Required. Storage account key.
Service = <data number>	Required. Storage Data Types (1:Blob,2:File,3:Table,4:Queue) ※Table/Queue is not supported
Protocol = [<protocol number>]	Protocol used for communication. (1:HTTPS, 2:HTTP, defaults to 1) ※For FileStorage, HTTP is not supported.
BlobEndpoint=[<Endpoint>]	The endpoint for connecting to BlobStorage on Azure IoT Edge. If not specified, it is connected to BlobStorage on Azure Portal. ※FileStorage is not supported.

2.2.2. CaoController::Execute method

Refer to 2.3.1 for the command names and details that can be used.2.3.1

SYNOPSIS 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 the method by which each provider creates a variable object. Only variables 2.4.1 can be used in variable names.2.4.1

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

<bstrVariableName> : [in] Variable name
 <bstrOption> : [in] Option string

The following options can be used in the option string:

Table 2-3 CaoController::AddVariable Option-String

Option	Meaning
Overwrite[=<True/False>]	Overwrite setting when there is a file at the copy destination. True: Overwrite. False: Do not overwrite. (default value)
Path[=<pathname>]	Describes the path to the file destination directory. (default: root directory of bucket) Use "/" (one-byte slash) as the path delimiter.

2.2.4. CaoController::AddFile method

This method creates a file object. Specifies as the name the file or directory that corresponds to the object. The file or directory specified here is the root directory of the tree structure formed by the file object. For this reason, you must optionally specify a path to the location of this file or directory. AddFile parameter specifications are shown below.

SYNOPSIS AddFile(<bstrObjectName:VT_BSTR>[,<bstrOption:VT_BSTR>])

<bstrObjectName> : [in] Object name
 <bstrOption> : [in] Option string

Table 2-4 CaoController::AddFile Option-String

Option	Description
FileName = <filename, directoryname>	Required. Specifies the file or directory name to be operated on. (*1)
Path[=<pathname>]	Describes the path to the directory containing the file specified as the file name. (default: root directory of bucket) Enter an absolute path. For relative paths, AddFile method fails. Use "/" (one-byte slash) as the path delimiter.
@Create[=<0-4>]	Creates a file according to this option value when there is no specified file. 0: No file is created. (default) 1: Create a file. (For BlobStorage, it is created as a blocking Blob.) 2: Create a directory. (Supported for FileStorage only) 3: Create a file. (For BlobStorage, it will be created as a paging Blob.) 4: Create a file. (For BlobStorage, create as additional Blob) This option is ignored if the specified file exists. If 0 is specified and the specified file or directory does not exist on Storage, the command fails. For FileStorage, any of 1, 3, or 4 is interpreted as a filecreation.

*1): The file or directory name can be any string except the following characters:

Table 2-5: List of characters that cannot be used

Option	Description
Characters that cannot be specified (Windows pass limit)	Symbol characters: Circle mark ("¥") Double quotes (" " ") To a greater or lesser extent("<>") Vertical bars/pipes (" ") COLON (":") Asterisk ("*") QUESTION MARK ("?") Circle mark/backslash ("¥") Slash ("/") ASCII control characters: "¥0,¥a,¥b,¥t,¥n,¥v,¥f,¥r" etc. Unicode control characters: "¥u0001~¥u0006 and ¥u000e~¥u001f", etc.

However, if a file/directory exists on a Storage that contains the following names, it is not recommended to use this provider to access Storage.

Table 2-6 List of deprecated characters and strings

Option	Description
Characters that cannot be specified (CAO Limitations)	Symbol characters: Parentheses ("{}") Square brackets ("[]") Parentheses ("()")
File name that cannot be specified (Restrictions on Windows Special Files)	CON, AUX, COM1,COM2,COM3,COM4,LPT1,LPT2,LPT3, PRN, NUL, etc. ※Including extensions (e.g. NUL.txt)
Generally deprecated characters	Tild ("~") Caret ("^") Sharp sign ("#") Backtick ("`")
File names that are not generally recommended	• The filename and the first character of the directory name are a period (" . ")

2.2.5. CaoController::get_FileNames Property

Retrieves a list of filenames that can be specified by AddFile method.

Table2-7 Option character string for CaoFile::GetFileNames

Option ¹	Description
Filter[=<filter-string>]	Filter settings for the retrieval list. (Default: "*")

2.2.6. CaoFile::AddFile method

Create a file object as described in 2.2.3 above. The name of a file or directory can only be a file in the directory that corresponds to CaoFile object that executes this method. For this reason, Path option, which is an option in 2.2.3 above, is ignored. This method will fail if CaoFile to be executed does not correspond to a directory.

2.2.7. CaoFile::Copy method

This method copies the file or directory to the specified location. The behavior varies depending on the type of file supported by the object, as shown in Table 2-7.

Table 2-7: Copy method operation for each supported file

Option	Description
File	Copy the file.
Directory	Recursively copies all directories, files within them, and subdirectories.

Copy parameter specifications are shown below.

SYNOPSIS AddFile(<bstrDest:VT_BSTR>[,<bstrOption:VT_BSTR>])

<bstrDest> : [in] File destination

<bstrOption> : [in] Option string

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

Table 2-8 CaoFile::Copy Option-String

Option ²	Description
Overwrite[=<True/False>]	Overwrite setting when there is a file at the copy destination. True: Overwrite. False: Do not overwrite. (default value)

The copy destination is specified as follows.

How to specify the destination of a copy of a 2-9 CaoFile::Copy and the destination of that copy

How to specify the copy destination	Results	Example: Specifying Method	Copy results
Absolute path	Copies to the specified destination location.	/TestDir2/Test2.txt	/TestDir2/Test2.txt
Relative path	Copy to a location relative to the directory where the source file is located.	./TestDir2/Test.txt	/TestDir/TestDir2/Test.txt
File name only	Copy to the same directory as the copy source.	Test2.txt	/TestDir/Test2.txt
Delimiter at the end of the path	Copy the file with the same name as the copy source to the copy destination directory.	/TestDir2/	/TestDir2/Test.txt

In this TestDir/Test, the source path is "/path.txt".

If any of the following conditions is entered in the copy destination, an error is returned.

- When the copy destination and copy source are the same.
- The copy destination is a subdirectory of the copy source.

² Square brackets ("[]") indicate optional characters.

2.2.8. CaoFile::Delete method

Deletes the file corresponding to the object. The object is not erased after the file is deleted, so you must erase it on the client when you do not need it.

※After calling this method, all operations of the corresponding CaoFile and CaoFile under that directory will fail.

2.2.9. CaoFile::Move method

Move method executes the 2.2.5CaoFile::Copy method described above, and then executes the 2.2.6CaoFile::Delete method.2.2.7CaoFile::Copy method2.2.8CaoFile::Delete method

The destination path specification and option-string are the same as for Copy method.

※After calling this method, all operations of the corresponding CaoFile and CaoFile under that directory will fail.

2.2.10. CaoFile::get_FileNames Property

Get the file name list in the directory corresponding to CaoFile object. This property will fail if the file corresponds to CaoFile object.

Table 2-10 CaoFile::GetFileNames Option-String

Option ³	Description
Filter[=<filter-string>]	Filter settings for the retrieval list. (Default: "*. * ")

2.2.11. CaoFile::get_Attribute Property

GetAttribute method determines whether the object is a file or a directory.

Attributes and their values that can be retrieved in the table 2-11 CaoFile::GetAttribute

Attribute	Value
Files (for FileStorage)	0x080
File (for BlobStorage (Block Blob))	0x080
File (for BlobStorage (additional Blob))	0x081
File (for BlobStorage (page Blob))	0x082
Directory	0x010

³ Square brackets ("[]") indicate optional characters.

2.2.12. CaoFile::get_Path property

Retrieves the full path to the parent directory where the file or directory corresponding to the object exists. The retrieved value does not include the file name.

※For/TestDir/Test.txt,/TestDir/is obtained.

2.2.13. CaoFile::get_Size

Gets the size of the file corresponding to the object. For a directory, the total number of files in the directory is obtained.

2.2.14. CaoFile::get_Type Property

Gets the extension of the file corresponding to the object. It will fail if it is a directory.

※'. Contains "" (period).

2.2.15. CaoFile::get_DateLastModified Property

Gets the last modified time of the file corresponding to the object. In the case of a directory, the last modification time of the file in the directory with the most recent modification time is obtained.

2.2.16. CaoFile::get_Value Property

Gets the contents of the file corresponding to the object. When the file extension is "txt", the content is obtained as a character string. Otherwise, the content is obtained as a binary array. It will fail if it is a directory.

2.2.17. CaoFile::put_Value Property

Set the contents of the file corresponding to the object. The client can set whether it sends data to the provider as a string or as a binary array. It will fail if it is a directory.

※For BlobStorage (additional Blob): Adds to a file.

※For BlobStorage (paged Blob): The data to be set cannot be accepted unless it is an integer multiple of 512 bytes.

2.2.18. CaoVariable::get_VariableNames Property

The variable of 2.4.1 is acquired.

2.2.19. CaoVariable::get_Value Property

Retrieves information corresponding to a variable. See 2.4.1 for the implementation status and retrieval data of each variable.2.4.1

2.2.20. CaoVariable::put_Value Property

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

2.3. Command list

2.3.1. CaoController classes

List of 2-12 CaoController::Execute Commands

Command	Function	
Upload	Upload a local file to Storage.	P. 1214
Download	Downloading Storage files locally.	P. 1215

Upload

Syntax	<i>Object. Upload(<Data>)</i>
Argument	<p><Data>= VT_VARIANT VT_ARRAY</p> <p>Array[0]: Full path of local file (VT_BSTR) (*1)</p> <p>Array[1]: Path of Storage file (VT_BSTR) (*1)</p> <p>Array[2]: Override setting when the same name file exists in the copy destination. (VT_BOOL)</p> <p>True: Overwrite.</p> <p>False: Do not overwrite. (Default value. Optional)</p>
Return Value	None
Description	<p>Upload the local file to Storage.</p> <p>For BlobStorage, upload it as a blocking Blob.</p> <p>Empty folders are not uploaded for BlobStorage.</p>

Download

Syntax	<i>Object.Download(<Data>)</i>
Argument	<Data>= VT_VARIANT VT_ARRAY Array[0]: Full path of Storage file (VT_BSTR) (*2) Array[1]: Full path of Local file (VT_BSTR) (*2) Array[2]: Override setting when the same name file exists in the copy destination. (VT_BOOL) True: Overwrite. False: Do not overwrite. (Default value. Optional)
Return Value	None
Description	Download Storage file to your local computer.

*1): The following shows an example of the operation specifications and path specification when uploading.

Table 2-13: List of path interpretation specifications for uploading

Pattern	Local files Path specification	In Storage Path specification	Upload Behavior	Remarks
1	Full path to the file	File name	Upload with the specified Storage filename	-
2	Full path to the file	Directory name	Upload to the specified Storage under the same name as the local file	Storage path is interpreted as a directory if it ends with a/(slash) when you specify it for uploading.
3	Full path to the folder	Directory name	Upload the specified local folder and all files and folders under the specified Storage directory.	If a local file is specified as a folder, it is interpreted as a directory with or without a trailing slash in Storage path.

Table 2-14: Example of path specification when uploading

Pattern	Example of Path Specification for Local Files	Sample Storage Path Specifications	Files generated during upload
1	C:\TestDir\Test.txt	/Test2.txt	/Test2.txt
	C:\TestDir\Test2.txt	/TestDir2	/TestDir2
2	C:\TestDir\Test.txt	/TestDir2/	/TestDir2/Test.txt
	C:\TestDir\Test2.txt	/TestDir/TestDir2/	/TestDir/TestDir2/Test2.txt
3	C:\TestDir	/TestDir2/	/TestDir2/TestDir/ File or folder under the above and
	C:\TestDir	/TestDir2	/TestDir2/TestDir/ File or folder under the above and

*2): The following is an example of the operation specification and path specification at the time of downloading.

Table 2-15: List of path interpretation specifications for downloading

Pattern	In Storage Path specification	Local files Path specification	Download Operations	Remarks
1	File name	File name	Storage file is downloaded to the specified local file path.	-
2	File name	Folder name	A file similar to Storage filename is downloaded directly under the specified local folder.	If the local path ends with a ¥ (circle mark) or a directory with the name specified in the local file path already exists, it is interpreted as a directory and Storage file specified under it is downloaded.
3	Folder name	Folder name	All Storage directories and files under them are downloaded directly under the specified local folder.	If Storage path is a directory, it is interpreted as a directory with or without a ¥ (circle mark) at the end of the local path, and the directory is generated and downloaded under it.

Table 2-16: Path specification examples for downloading

Pattern	Pathing a Storage	Path Specification for Local Files	Files Generated When Downloading
1	/TestDir/Test.txt	C:¥TestDir¥Test.txt	C:¥TestDir¥Test.txt
	/TestDir/Test2.txt	C:¥TestDir¥Test ※ If a folder named Test does not exist	C:¥TestDir¥Test
2	/TestDir/Test.txt	C:¥TestDir¥	C:¥TestDir¥Test.txt
	/TestDir/Test2.txt	C:¥TestDir¥Test ※ If a folder named Test already exists	C:¥TestDir¥Test¥Test2.txt
3	/TestDir/	C:¥TestDir¥	C:¥TestDir¥TestDir Files or directories under the above and/TestDir/
	/TestDir2	C:¥TestDir2	C:¥TestDir2¥TestDir2 Files or directories under the above and/TestDir2
	/TestDir3	C:¥	C:¥TestDir3 Files or directories under the above and/or TestDir3

2.4. Variable list

2.4.1. CaoController classes

Table 2-17 CaoController Class User Variable List

Variable name	Data Type	Description	Attribute		Option	
			Get	Put	Overwrite	Path
*	VT_BSTR VT_ARRAY	<p>Send a file to Azure Storage's Blob Storage in synchronous mode.</p> <p><Source file path> [<destination file name>]</p> <p>※ The destination file name is optional. If the destination file name is omitted, the file is saved as the name of the source file.</p> <p>※ The data is saved in the folders specified by Path of Azure Storage.</p> <p>※ If there is a file with the same name on Azure Storage, it will be overwritten if OverWrite is True.</p>	-	○	○	○

Table 2-18 CaoController Class System Variable List

Variable name	Data Type	Description	Attribute	
			Get	Put
@AZURESTORAGESDK_VERSION	VT_BSTR	Version of Microsoft Azure Storage SDK for.NET.	○	-