

# Amazon Web Service S3 provider

Version 1.3.1

User's Guide

December 13, 2021

[Notes]

**[Revision History]**

Version	Date	Description
1.0.0	2017-07-06	First edition
1.0.1	2017-07-20	Add CaoFile::get_Value,CaoFile::put_Value Document Modification
1.0.2	2017-09-28	Add CaoController::Get_FileNames
1.1.0	2020-03-02	Add CaoController::AddVariable
1.2.0	2021-06-02	Supports TLS 1.2. Change to EXE format.
	2021-07-08	Adjusting the appearance of the user's guide.
1.3.0	2021-09-13	Add item to CaoWorkspace::AddController options
1.3.1	2021-12-13	Bug Fix. Document modification.

**[Supported Devices]**

Model	Version	POINTS OF CAUTION

## 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 .....	8
2.2.3. CaoController::AddVariable method.....	8
2.2.4. CaoController::AddFile method.....	9
2.2.5. CaoController::get_FileNames Property .....	11
2.2.6. CaoFile::AddFile method .....	11
2.2.7. CaoFile::Copy method .....	11
2.2.8. CaoFile::Delete method.....	13
2.2.9. CaoFile::Move method.....	13
2.2.10. CaoFile::get_FileNames Property .....	13
2.2.11. CaoFile::get_Attribute Property .....	13
2.2.12. CaoFile::get_Path Property .....	14
2.2.13. CaoFile::get_Size property .....	14
2.2.14. CaoFile::get_Type Property.....	14
2.2.15. CaoFile::get_DateLastModified Property .....	14
2.2.16. CaoFile::get_Value Property .....	14
2.2.17. CaoFile::put_Value Property .....	14
2.2.18. CaoVariable::get_VariableNames Property .....	14
2.2.19. CaoVariable::get_Value Property .....	14
2.2.20. CaoVariable::put_Value Property .....	14
2.3. Command list .....	15
2.3.1. CaoController classes .....	15
2.4. List of Variables.....	19
2.4.1. CaoController classes .....	19

## 1. Introduction

This document is the user's guide of the CAO provider who is responsible for sending and receiving data to and from Amazon Web Service storage service S3.

The CAO provider (CaoProvAmazonAWSS3. exe) described in this document is called AWSS3 provider. Section 2 provides an outline of AWSS3 providers and more information about variables.

It uses AWS SDK for.NET to communicate with AWSS3.

For more information about these, see:

Net Framework 4.6.2 is required to use this provider.

AWS SDK for.NET Site Links

- AWS SDK for .NET

<https://github.com/aws/aws-sdk-net/> ([Link](https://github.com/aws/aws-sdk-net/))<https://github.com/aws/aws-sdk-net/>

AWS SDK for.NET Copyrights and Licenses

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

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

## 2. Provider Overview

### 2.1. Introduction

AWSS3 providers are CAOs that communicate with S3 using AWS SDK for.NET. Its file format is EXE, which is loaded dynamically from the CAO engine when used. To use AWSS3 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 AWSS3 Providers**

File name	CaoProvAmazonAWSS3.exe
ProgID	CaoProv.Amazon.AWSS3
Registry registration	RegistAsm.bat CaoProvAmazonAWSS3.exe
Deletion of Registry Registration	UnregistAsm.bat CaoProvAmazonAWSS3.exe

## 2.2. Method Properties

### 2.2.1. CaoWorkspace::AddController method

AWSS3 providers connect to AWSS3 by referring to the connection parameters for communication at the time of AddController.

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

BstrCtrlName : [in] Controller name  
 BstrProvName : [in] Provider name. Fixed value = "CaoProv.Amazon.AWSS3"  
 BstrPcName : [in] The name of the provider's running machine  
 BstrOption : [in] option string

**Table 2-2 Optional character string of CaoWorkspace::AddController**

OPTIONS	Description
AccessKeyID = <access key ID>	Specify the user's access key ID to connect to mandatory.S3.
SecretAccessKey =<Secret Access Key>	Required.Specifies the user's secret access key to connect to S3.
Bucket = <bucket name>	Required. The S3 bucket name.
Protocol = [<protocol-number>]	The protocol used for communication. (1:HTTPS, 2:HTTP, Default: 1)
Endpoint=[<Endpoint>]	The URL of the endpoint used to execute the request. When EndPoint and RegionEndpoint are specified, RegionEndpoint is used.
RegionEndpoint=[<Region>]	Regional endpoint that you can use to make your requests. (default: ap-northeast-1)

The following are used for RegionEndpoint strings.

**Table 2-3 RegionEndpoint strings of CaoWorkspace::AddController option**

String	Destination
af-south-1	Africa (Cape Town)
ap-east-1	Asia Pacific (Hong Kong)
ap-northeast-1	Asia Pacific (Tokyo)
ap-northeast-2	Asia Pacific (Seoul)
ap-northeast-3	Asia Pacific (Osaka)
ap-south-1	Asia Pacific (Mumbai)
ap-southeast-1	Asia Pacific (Singapore)
ap-southeast-2	Asia Pacific (Sydney)
ca-central-1	Canada (Central)
cn-north-1	China (Beijing)
cn-northwest-1	China (Ningxia)
eu-central-1	Europe (Frankfurt)
eu-north-1	Europe (Stockholm)
eu-south-1	Europe (Milan)
eu-west-1	Europe (Ireland)
eu-west-2	Europe (London)
eu-west-3	Europe (Paris)
me-south-1	Middle East (Bahrain)
sa-east-1	South America (Sao Paulo)
us-east-1	US East (N. Virginia)
us-east-2	US East (Ohio)
us-west-1	US West (N. California)
us-west-2	US West (Oregon)

### 2.2.2. CaoController::Execute method

See section 2.3.1 for the command names and details.

**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

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 name  
 <bstrOption> : [in] option string

The following are used for an optional character string.

**Table 2-4 Optional character string of CaoWorkspace::AddVariable**

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 the name of the file or directory that you want the object to correspond to. 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 that points to the position of this file or directory. AddFile argument specifications are shown below.

**Format** AddFile(<bstrObjectName:VT\_BSTR>[,<bstrOption:VT\_BSTR>])

<bstrObjectName> : [in] Object name

<bstrOption> : [in] Option string

**Table 2-5 Optional character string of CaoController::AddFile**

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 where the file specified as the file name is located. (Default: root directory of the bucket) Enter an absolute path. For relative paths, AddFile method fails. Use "/" (single-byte slash) as the path delimiter.
@Create[=<0-2>]	If the specified file does not exist, the file is created according to this option value. 0: Do not create files. (default) 1: Create the file. 2: Create a directory. This option is ignored if the specified file exists. If 0 is specified and the specified files or directories do not exist on AWSS3, the operation fails.

\*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 (Restrict Windows Paths)	Symbol characters: Yen mark ("¥") Double quotation marks (" ") To a greater or lesser extent("<>") Vertical bar/pipe (" ") COLON (":") Asterisk ("*") QUESTION MARK ("?") Yen mark/backslash ("¥") Slash ("/") ASCII control characters: ¥0, ¥a, ¥b, ¥t, ¥n, ¥v, ¥f, ¥r, and so on. 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 the extension (e.g. NUL.txt)
Generally deprecated characters	Tilde ("~") Caret ("^") Pound sign ("#") Backtick ("`")
Generally deprecated file name	File name and the first character of the directory name is a period (".")

### 2.2.5. CaoController::get\_FileNames Property

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

**Table2-6 Option character string for CaoFile::GetFileNames**

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

### 2.2.6. CaoFile::AddFile method

Create a file object as in 2.2.2 above. The name of a file or directory can only be a file in the directory that corresponds to CaoFile objects that execute this method. For this reason, Path option described in 2.2.2 above is ignored. This method fails if CaoFile objects to be executed do not correspond to directories.

### 2.2.7. CaoFile::Copy method

This method copies the file or directory to the specified location. The behavior changes as shown in Table 2-5, depending on the type of file supported by the object. Table 2-7 Behavior of Copy Methods for Each Supported File

**Table 2-7 Behavior of Copy Methods for Each Supported File**

Option	Description
FILES	Copy the file.
Directories	Recursively copy a directory and all of its files and subdirectories.

Copy argument specifications are shown below.

**Format** AddFile(<bstrDest:VT\_BSTR>[,<bstrOption:VT\_BSTR>])

<bstrDest> : [in] where to copy the files

<bstrOption> : [in] option string

**Table 2-8 Optional character string of CaoFile::Copy**

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

The location to copy using the copy destination specification method is as follows.

**Table 2-9 CaoFile::Copy copy destination and its copy destination**

How to specify the copy destination	Result	Example: Specifying method	Copy result
Absolute path	Copies to the specified 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 end of path	Copy the file with the same name as the copy source to the copy destination directory.	/TestDir2/	/TestDir2/Test.txt

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

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

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

### 2.2.8. CaoFile::Delete method

Delete the file corresponding to the object.

After you delete a file, the object is not erased, so you must clear the object on the client when you don't need it.

※After this method is called, any operations on CaoFile or CaoFile under that directory will fail.

### 2.2.9. CaoFile::Move method

Move method executes 2.2.4CaoFile::Copy method and then 2.2.5CaoFile::Delete method.2.2.7CaoFile::Copy method2.2.8CaoFile::Delete method

In this case, the destination path specification method and option character strings are the same as for Copy method.

※After this method is called, any operations on CaoFile or CaoFile under that directory will fail.

### 2.2.10. CaoFile::get\_FileNames Property

If CaoFile objects correspond to directories, get a list of filenames and directory names immediately below them. A delimiter ("/") is added to the end of the directory name.

It will fail if CaoFile objects correspond to files.

**Table 2-10 Optional character string of CaoFile::GetFileNames**

OPTIONS	Description
Filter[=<filter-string>]	Configure filters for the retrieve list. (Default: "*.*") ※The following wildcard characters are allowed: *: Any length of characters (zero or more characters) ?: Any single character

### 2.2.11. CaoFile::get\_Attribute Property

GetAttribute method determines whether objects correspond to files or directories.

**Tables 2-11 Attributes that can be obtained and their values with CaoFile :: GetAttribute**

Attribute	Value
FILES	0x080
Directories	0x010

**2.2.12. CaoFile::get\_Path Property**

Get the full path of the file or directory corresponding to the object.

**2.2.13. CaoFile::get\_Size property**

Get the size of the file corresponding to an object. In the case of a directory, the sum of the files in the directory is obtained.

**2.2.14. CaoFile::get\_Type Property**

Get the extension of the file corresponding to the object. It will fail for directories.

※. Contains " (period).

**2.2.15. CaoFile::get\_DateLastModified Property**

Get the last modification time of the file corresponding to the object. For a directory, retrieves the date and time of the last modification of the file in the directory that most recently appeared.

**2.2.16. CaoFile::get\_Value Property**

Get the contents of the file corresponding to the object. If the file extension is "txt", the content is retrieved as a string. Otherwise, the content is retrieved as a binary array. The directory fails.

**2.2.17. CaoFile::put\_Value Property**

Sets the content of the file corresponding to the object. The client can set whether to send data to the provider as a string or as a binary array. The directory fails.

**2.2.18. CaoVariable::get\_VariableNames Property**

The variable of 2.4.1 is acquired.

**2.2.19. CaoVariable::get\_Value Property**

Get information corresponding to a variable. Refer to 2.4.1 for the implementation status of each variable and the acquired data.

**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 the local file to S3.	P. 15
Download	The file of S 3 is downloaded locally.	P. 16

## Upload

<b>Syntax</b>	<i>Object. Upload(&lt;Data&gt;)</i>
<b>Argument</b>	<p>&lt;Data&gt;= VT_VARIANT VT_ARRAY</p> <p>Array[0]: Full path of local file (VT_BSTR)(*1)</p> <p>Array[1]: Full path of the S3 file (VT_BSTR)(*1)</p> <p>Array [2]: Overwrite setting when there is a file with the same name in the destination. (VT_BOOL)</p> <p>True: Overwrite.</p> <p>False: Do not overwrite. (Default value. (can be omitted))</p>
<b>Return Values</b>	None
<b>Description</b>	Upload the local file to S3.

## Download

---

<b>Syntax</b>	<i>Object. Download(&lt;Data&gt;)</i>
<b>Argument</b>	<b>&lt;Data&gt;= VT_VARIANT VT_ARRAY</b> Array[0]: Full path of the S3 file (VT_BSTR)(*2) Array [1]: Full path of the local file (VT_BSTR)(*2) Array [2]: Overwrite setting when there is a file with the same name in the destination. (VT_BOOL) True: Overwrite. False: Do not overwrite. (Default value. (can be omitted))
<b>Return Values</b>	None
<b>Description</b>	Download the S3 file to your local PC.

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

**Table 2-13 List of path interpretation specifications when uploading**

Pattern	Local files Path specification	For S3 Path specification	Upload Behavior	NOTES
1	Full path of the file	File name	Upload with specified S3 file name	-
2	Full path of the file	Directory name	Local to the specified S3 directory Upload with the same name as the file	The S3 path is interpreted as a directory if it ends with a/(slash) in the upload specification.
3	Full path of the folder	Directory name	Upload all the specified local folders and files and folders under the specified S3 directory.	If the local file specification is a folder, it is interpreted as a directory, regardless of whether a slash is present at the end of the S3 path.

**Table 2-14 Example path specifications when uploading**

Pattern	Example of specifying a path to a local file	Example of S3 Path Specification	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 operation specifications and path specification when downloading.

**Table 2-15 List of path interpretation specifications when downloading**

Pattern	For S3 Path specification	Local files Path specification	Download Operations	NOTES
1	File name	File name	The S3 file is downloaded to the specified local file path.	-
2	File name	Folder name	The same file as the file name of S3 will be downloaded directly under the specified local folder.	If the local path ends with a backslash (¥) or a directory with the name specified in the local file path already exists, it is interpreted as a directory and the specified S3 file under that directory is downloaded.
3	Folder name	Folder name	The S3 directory and all files under it are downloaded directly under the specified local folder.	If the path of S3 is a directory, it is interpreted as a directory with or without the backslash (¥) at the end of the local path, and the directory is created and downloaded under it.

**Table 2-16 Example path specifications during download**

Pattern	S3 path specification	Path Specification for Local Files	Files Generated During Download
1	/TestDir/Test.txt	C:¥TestDir¥Test.txt	C:¥TestDir¥Test.txt
	/TestDir/Test2.txt	C:¥TestDir¥Test ※ If Test folder 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 called Test already exists	C:¥TestDir¥Test¥Test2.txt
3	/TestDir/	C:¥TestDir¥	C:¥TestDir¥TestDir Files or directories under the above and/or TestDir/
	/TestDir2	C:¥TestDir2	C:¥TestDir2¥TestDir2 Files or directories under the above and/or TestDir2
	/TestDir3	C:¥	C:¥TestDir3 Files or directories under the above and/or TestDir3

## 2.4. List of Variables

### 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 S3 in synchronous mode.</p> <p>&lt;Source file path&gt; [&lt;destination file name&gt;]</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 S3.</p> <p>※ If there is a file with the same name on S3, it will be overwritten if OverWrite is True.</p>	-	○	○	○

**Table List of 2-18 CaoController Class System Variables**

Variable name	Data Type	Description	Attribute	
			Get	Put
@AWSSDK_VERSION	VT_BSTR	AWSSDK versions for.NET.	○	-