Automation 360 – AWS S3 File management Bot

Readme

Version 2.0

06/08/2023

Table of Contents

1. Introduction

This document contains all essential information for the user to make full use of this AWS S3 File

Management Bot. It includes the description of the functions and capabilities and step-by-step procedure

for setup & configuration of the bot.

1.1 Overview

AWS S3 File Management Bot can be used to upload and download files to and from your AWS S3 bucket.

1.2 Use cases

The key use cases include:

• File Management in AWS S3 Bucket

- Upload file to AWS S3 Bucket from the local system
- Download files from AWS S3 bucket to the local system
- Delete files from AWS S3 Bucket

2. Requirements & Prerequisites

2.1 System Requirements

Below are the minimum system requirements for running bots as an Enterprise A2019 (Cloud deployed) or Community Edition user on your local machine.

Hardware Requirements

Device	Processor	RAM	Storage (Free Space)	Network
Machine	Intel Core i5 2.6 GHz	4 GB minimum 8 GB recommended	32 GB	1 GbE

Bot Creator and Bot Runner	No additions to the machine requirements	No additions to the machine requirements	Add 100 through 150 KB per Automation Anywhere script Add 40 through 50 GB per long-term project	No additions to the machine requirements

Browser Requirements

The user interface for Automation Anywhere Enterprise is through a browser. Login to your device then login to Enterprise Control Room through a browser.

Browser	Browser Version	Automation Anywhere plug-in version[2]	Supported bot functions by Bot agent version		
Google Chrome[1]	57 or later	11 or 12	2.0.2	1.0.2	1.0.1
Microsoft Internet Explorer	11	N/A	None	Debugger only	All except Credential Vault

(1) Google Chrome re-verification

> **CAUTION:** Google Chrome requires re-verification of permissions when the Automation Anywhere Google Chrome extension (Version 11.3.3 or later) is updated. If prompted, click **Enable this item** in the Google Chrome message. Alternatively, re-enable the extension through chrome web store. Similarly, if you are deploying your Bot Runners from a master image, accept the permission from within that image.

(2) Google Chrome plug-in versions

> A2019 supports Chrome plug-in version 11. If you have other versions installed, you might not be able to record tasks using the **Record** feature or **Capture** action. To resolve this issue, do the following:
>
> 1. Rename the automation.chrome.agent registry key to automation.chrome.agent.old. The registry key is available at: Computer\HKEY_CURRENT_USER\Software\Google\Chrome\NativeMessagingHosts
> 2. Disable all other Google Chrome plug-in versions you have installed, except for version 11.

2.2 Prerequisites

- Automation 360 setup to run on your machine.
- .Net Framework 4.6.1 or above installed.
- Active AWS Account

3. Getting Started

3.1 Quick Start

3.1.1 Setup

- Install the Bot from Bot Store into your Control Room
- Navigate to AWS S3 File Management – Automation Anywhere in the Bot Store folder to examine the installed bot.

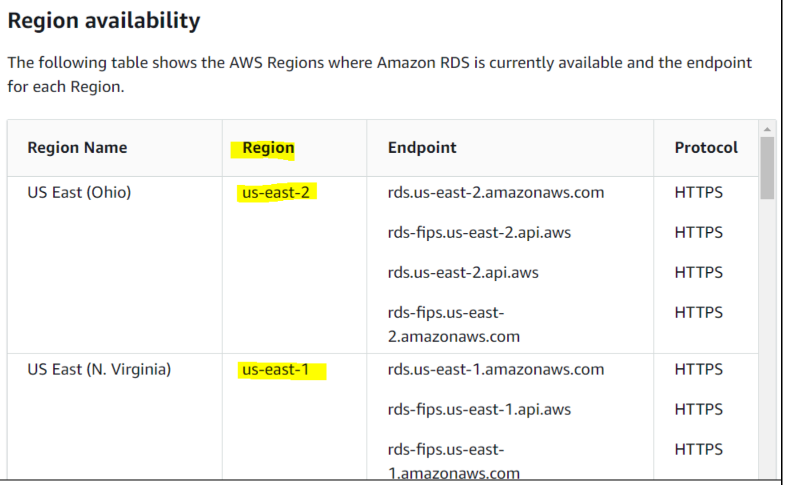
3.1.2 Configuration and Use

Create a credential locker with the name AWSLocker and create credentials with the AWSCredential. Add Attributes with the names AccessKey and SecretKey.

S/N	Variable Name	Variable Type	Function
1	BucketName	String	Name of the S3 Bucket to be used for upload and download operations. Example – testawsrpa
2	UploadFileName	String	This is the name of the file you wish to upload. Example – abc.pdf
3	UploadFilePath	String	This is the local system file path from which you would like to upload the file to AWS S3 Bucket. Example – “C:/users/user/test.pdf”
4	DownloadFileName	String	This is the name of the file you wish to download. Example – abc.pdf
5	DownloadFilePath	String	This is the local system file path to which you would like to download the file to AWS S3 Bucket. Example – “C:/users/user/test.pdf”
6	AccessKeyID	Credential	This is the Access key that provides access to AWS S3 Bucket and needs to be stored in Credential Vault. Information on how to generate

<table>
<tr><td></td><td></td><td></td><td>the keys can be found at this link:
https://docs.aws.amazon.com/general/latest/gr/managingaws-access-keys.html</td></tr>
<tr><td>7</td><td>SecretKey</td><td>Credential</td><td>This is the Secret Access key that provides access to AWS S3 Bucket and needs to be stored in Credential Vault.</td></tr>
<tr><td>8</td><td>CreateBucket</td><td>String</td><td>Enter the value as True if you want a new bucket to be created in S3. Enter 'False' if you don't need a new bucket to be created in S3.</td></tr>
<tr><td>9</td><td>Region</td><td>String</td><td>Use below Amazon documentation link to get the region system name
("Region" column in the table)
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.RegionsAndAvailabilityZones.html#Concepts.RegionsAndAvailabilityZones.Regions

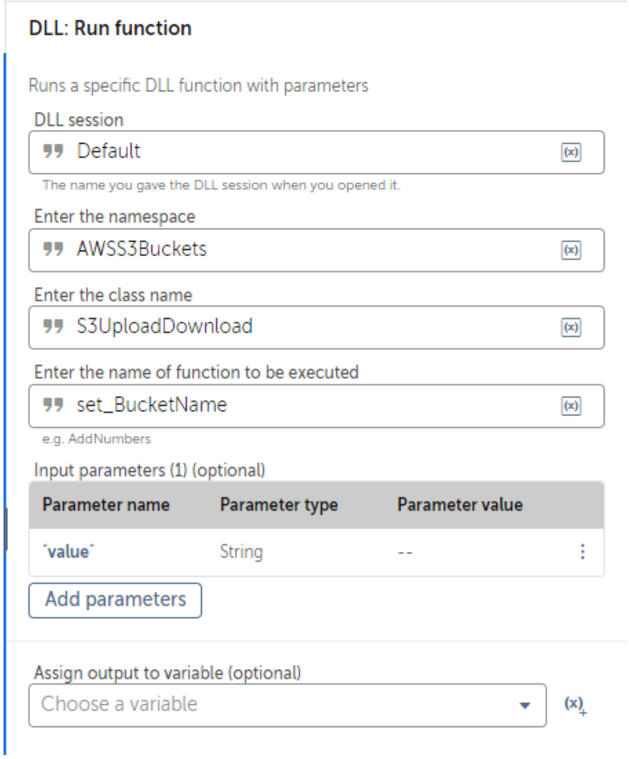
It supports all the regions supported by Amazon s3.


<b>Region availability</b>
The following table shows the AWS Regions where Amazon RDS is currently available and the endpoint for each Region.
<table>
<tr><th>Region Name</th><th>Region</th><th>Endpoint</th><th>Protocol</th></tr>
<tr><td>US East (Ohio)</td><td>us-east-2</td><td>rds.us-east-2.amazonaws.com</td><td>HTTPS</td></tr>
<tr><td></td><td></td><td>rds-fips.us-east-2.api.aws</td><td>HTTPS</td></tr>
<tr><td></td><td></td><td>rds.us-east-2.api.aws</td><td>HTTPS</td></tr>
<tr><td></td><td></td><td>rds-fips.us-east-2.amazonaws.com</td><td>HTTPS</td></tr>
<tr><td>US East (N. Virginia)</td><td>us-east-1</td><td>rds.us-east-1.amazonaws.com</td><td>HTTPS</td></tr>
<tr><td></td><td></td><td>rds-fips.us-east-1.api.aws</td><td>HTTPS</td></tr>
<tr><td></td><td></td><td>rds-fips.us-east-1.amazonaws.com</td><td>HTTPS</td></tr>
</table></td></tr>
<tr><td>10</td><td>DeleteFileName</td><td>String</td><td>This is the name of the file you wish to delete. Example – abc.pdf</td></tr>
</table>

Configure AccessKeyID and SecretKey to get the values from the Credential vault.
Add the values for other required variables as shown in the screenshots below for your reference

You can download the DLL source code from
https://github.com/partnersolutiondesk/RPA/tree/main/C%23%20DLLs/AWSS3-File%20Management

Upload Files to S3



DLL: Run function

Runs a specific DLL function with parameters

DLL session: Default
The name you gave the DLL session when you opened it.

Enter the namespace: AWSS3Buckets

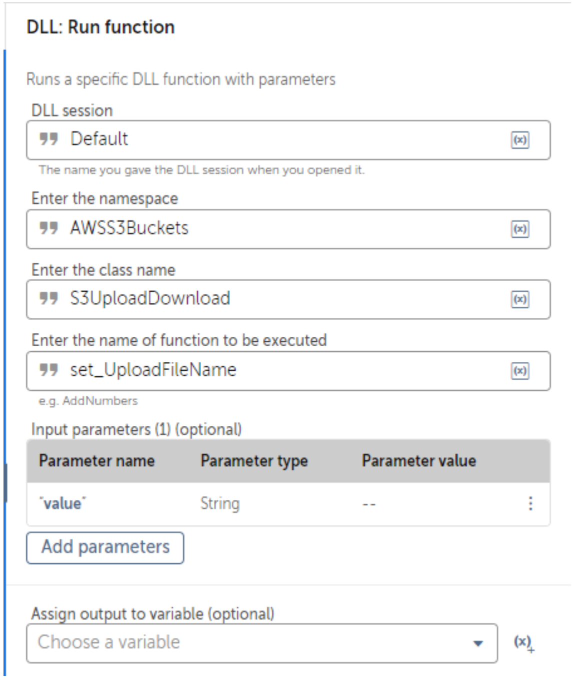
Enter the class name: S3UploadDownload

Enter the name of function to be executed: set_BucketName
e.g. AddNumbers

Input parameters (1) (optional)

Parameter name	Parameter type	Parameter value
"value"	String	--

Assign output to variable (optional): Choose a variable



DLL: Run function

Runs a specific DLL function with parameters

DLL session: Default

The name you gave the DLL session when you opened it.

Enter the namespace: AWSS3Buckets

Enter the class name: S3UploadDownload

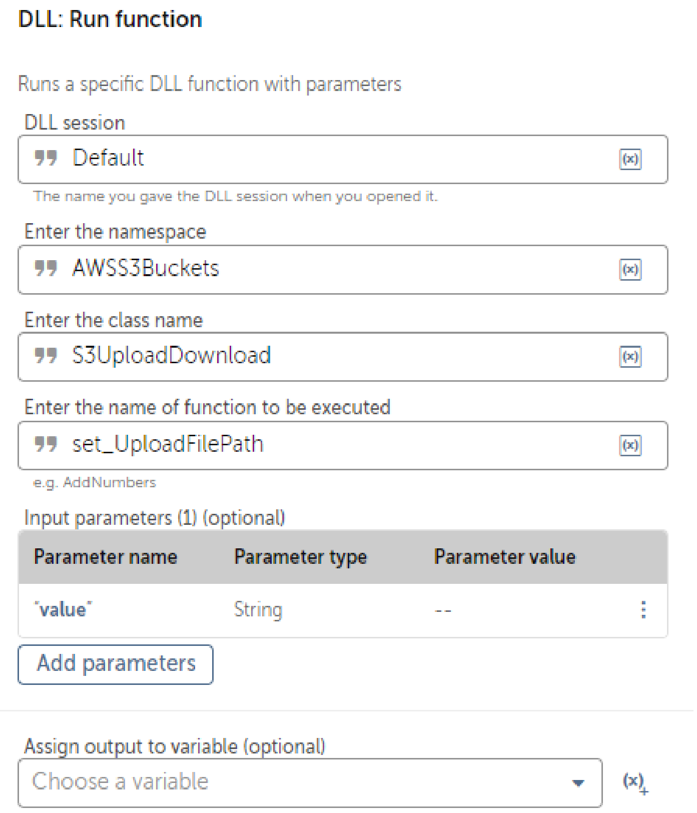
Enter the name of function to be executed: set_UploadFileName

e.g. AddNumbers

Input parameters (1) (optional)

Parameter name	Parameter type	Parameter value
"value"	String	--

Assign output to variable (optional): Choose a variable



DLL: Run function

Runs a specific DLL function with parameters

DLL session

Default

The name you gave the DLL session when you opened it.

Enter the namespace

AWSS3Buckets

Enter the class name

S3UploadDownload

Enter the name of function to be executed

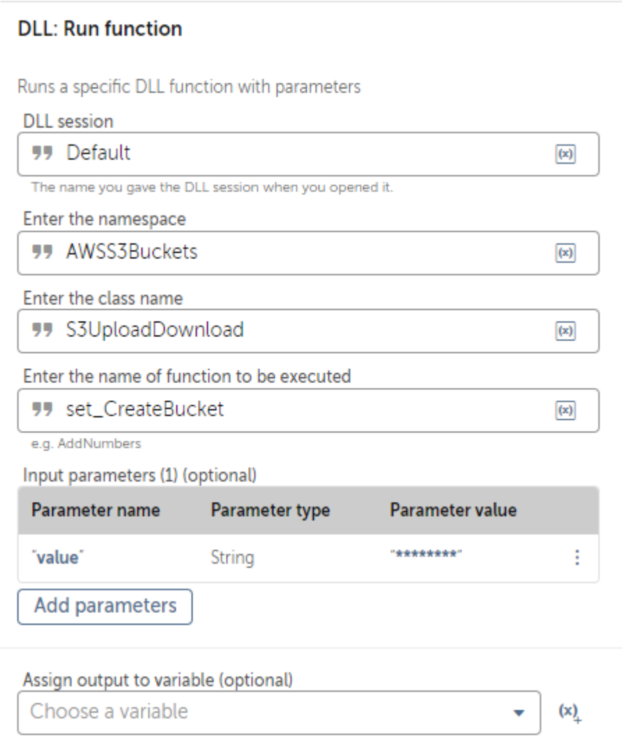
set_UploadFilePath

e.g. AddNumbers

Input parameters (1) (optional)

Parameter name	Parameter type	Parameter value
"value"	String	--

Assign output to variable (optional)



DLL: Run function

Runs a specific DLL function with parameters

DLL session

Default

The name you gave the DLL session when you opened it.

Enter the namespace

AWSS3Buckets

Enter the class name

S3UploadDownload

Enter the name of function to be executed

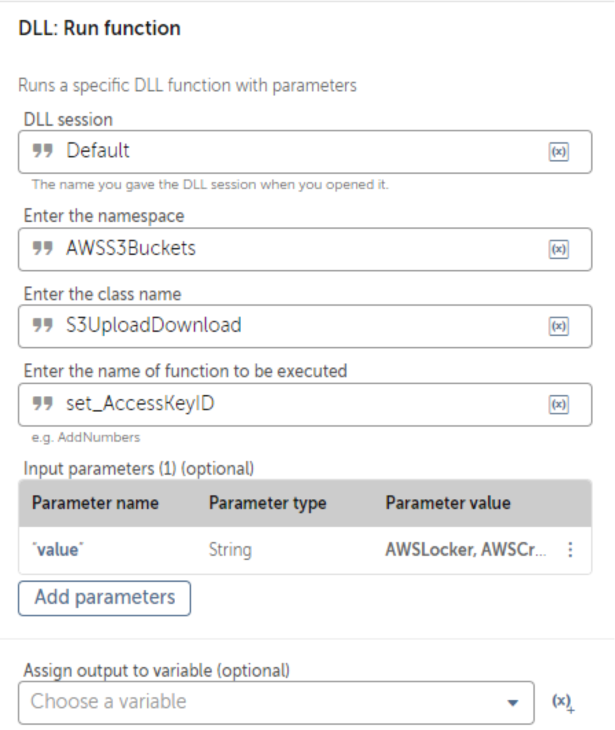
set_CreateBucket

e.g. AddNumbers

Input parameters (1) (optional)

Parameter name	Parameter type	Parameter value
"value"	String	"*********"

Assign output to variable (optional)



DLL: Run function

Runs a specific DLL function with parameters

DLL session: Default
The name you gave the DLL session when you opened it.

Enter the namespace: AWSS3Buckets

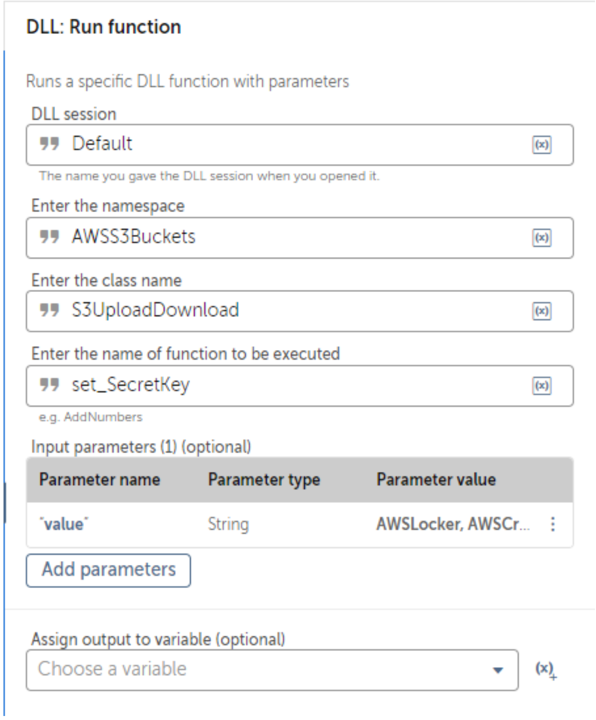
Enter the class name: S3UploadDownload

Enter the name of function to be executed: set_AccessKeyID
e.g. AddNumbers

Input parameters (1) (optional)

Parameter name	Parameter type	Parameter value
"value"	String	AWSLocker, AWSCr...

Assign output to variable (optional): Choose a variable



DLL: Run function

Runs a specific DLL function with parameters

DLL session

Default

The name you gave the DLL session when you opened it.

Enter the namespace

AWSS3Buckets

Enter the class name

S3UploadDownload

Enter the name of function to be executed

set_SecretKey

e.g. AddNumbers

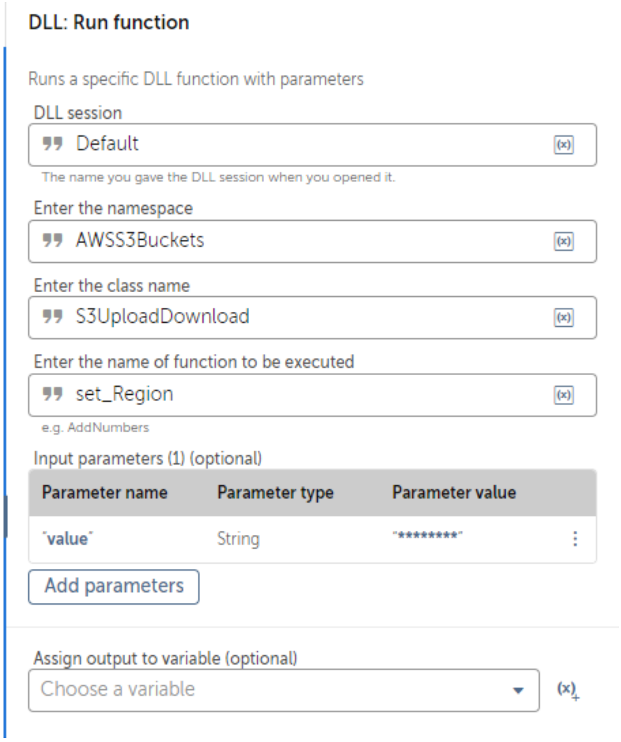
Input parameters (1) (optional)

Parameter name	Parameter type	Parameter value
`value`	String	AWSLocker, AWSCr...

Add parameters

Assign output to variable (optional)

Choose a variable



DLL: Run function

Runs a specific DLL function with parameters

DLL session

Default

The name you gave the DLL session when you opened it.

Enter the namespace

AWSS3Buckets

Enter the class name

S3UploadDownload

Enter the name of function to be executed

set_Region

e.g. AddNumbers

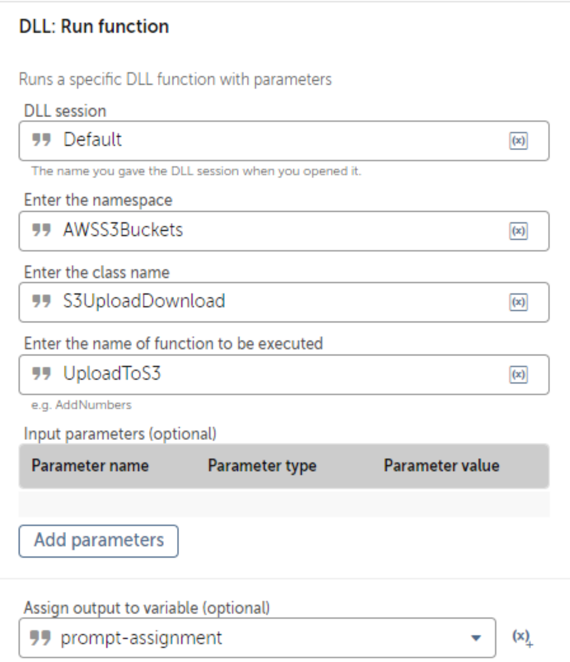
Input parameters (1) (optional)

Parameter name	Parameter type	Parameter value
"value"	String	"*********"

Add parameters

Assign output to variable (optional)

Choose a variable



DLL: Run function

Runs a specific DLL function with parameters

DLL session: Default

The name you gave the DLL session when you opened it.

Enter the namespace: AWSS3Buckets

Enter the class name: S3UploadDownload

Enter the name of function to be executed: UploadToS3

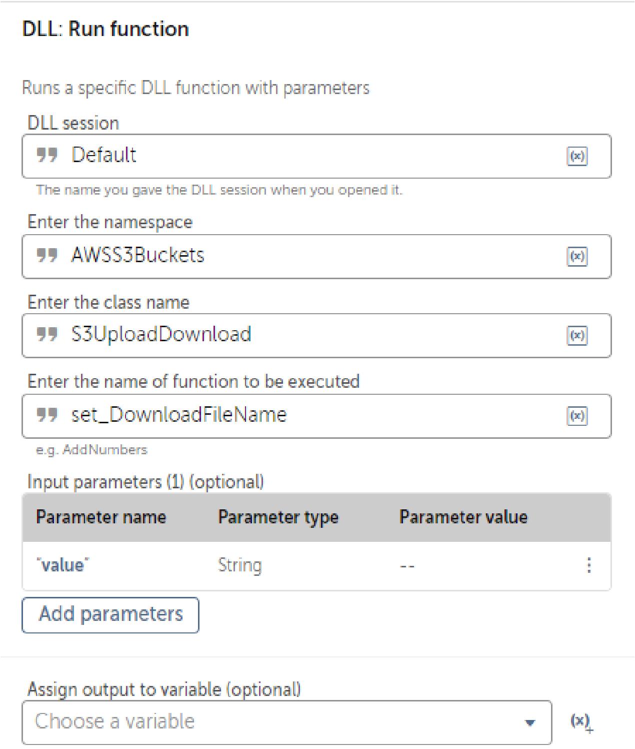
e.g. AddNumbers

Input parameters (optional)

Parameter name	Parameter type	Parameter value

Assign output to variable (optional): prompt-assignment

Download Files from S3



DLL: Run function

Runs a specific DLL function with parameters

DLL session

Default

The name you gave the DLL session when you opened it.

Enter the namespace

AWSS3Buckets

Enter the class name

S3UploadDownload

Enter the name of function to be executed

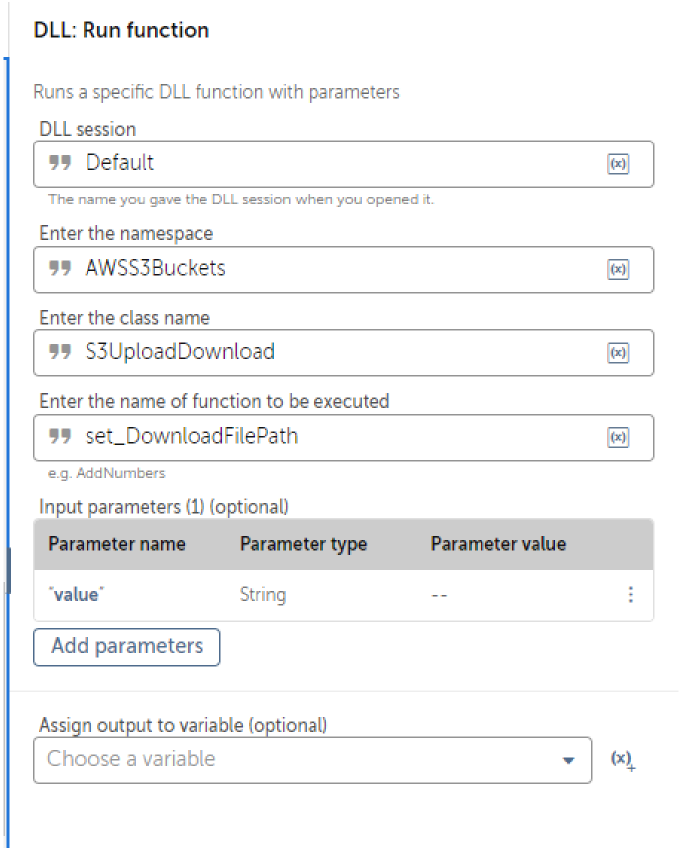
set_DownloadFileName

e.g. AddNumbers

Input parameters (1) (optional)

Parameter name	Parameter type	Parameter value
"value"	String	--

Assign output to variable (optional)



DLL: Run function

Runs a specific DLL function with parameters

DLL session: Default

The name you gave the DLL session when you opened it.

Enter the namespace: AWSS3Buckets

Enter the class name: S3UploadDownload

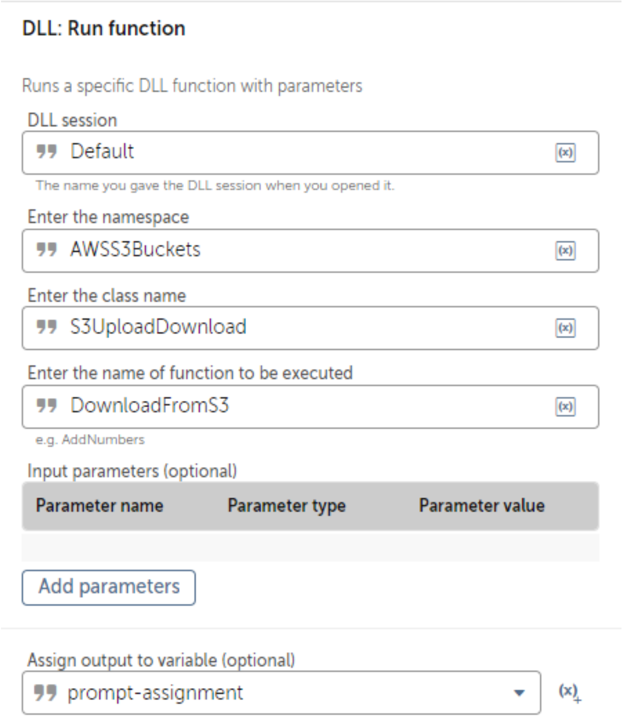
Enter the name of function to be executed: set_DownloadFilePath

e.g. AddNumbers

Input parameters (1) (optional)

Parameter name	Parameter type	Parameter value
"value"	String	--

Assign output to variable (optional): Choose a variable



DLL: Run function

Runs a specific DLL function with parameters

DLL session: Default

The name you gave the DLL session when you opened it.

Enter the namespace: AWSS3Buckets

Enter the class name: S3UploadDownload

Enter the name of function to be executed: DownloadFromS3

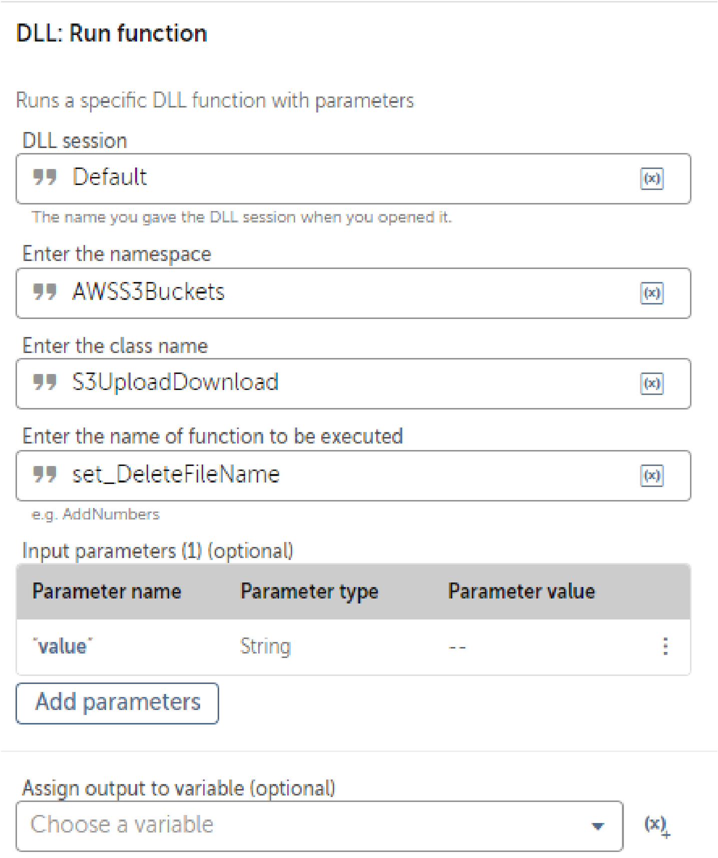
e.g. AddNumbers

Input parameters (optional)

Parameter name	Parameter type	Parameter value

Assign output to variable (optional): prompt-assignment

Delete Files from S3



DLL: Run function

Runs a specific DLL function with parameters

DLL session

Default

The name you gave the DLL session when you opened it.

Enter the namespace

AWSS3Buckets

Enter the class name

S3UploadDownload

Enter the name of function to be executed

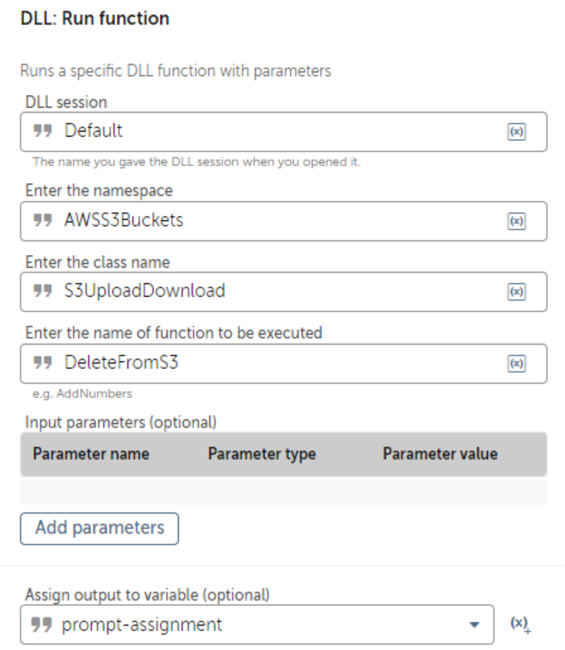
set_DeleteFileName

e.g. AddNumbers

Input parameters (1) (optional)

Parameter name	Parameter type	Parameter value
"value"	String	--

Assign output to variable (optional)



DLL: Run function

Runs a specific DLL function with parameters

DLL session

Default

The name you gave the DLL session when you opened it.

Enter the namespace

AWSS3Buckets

Enter the class name

S3UploadDownload

Enter the name of function to be executed

DeleteFromS3

e.g. AddNumbers

Input parameters (optional)

Parameter name	Parameter type	Parameter value

Assign output to variable (optional)

prompt-assignment

4. Support & FAQs

4.1 Support

Free bots are not officially supported. You can get access to Community Support through the following channels:

- You can get access to Community Support, connecting with other Automation Anywhere customers and developers on APeople – the Bot Building Forum, the Bot Store Support Forum, or the Developers Everywhere Group.
- Automation Anywhere also provides a Product Documentation portal which can be accessed for more information about our products and guidance on Enterprise A2019.

4.2 FAQs

Q: Can I make changes to this bot?

A: Absolutely. Free bots are for you to make use of, customize, and/or include in your builds as you see fit.

Q: Can I upgrade the packages for this bot?

A: Yes – but know that the bot was created/tested on a specific package version. Any modifications to that may result in unexpected outcomes – so make sure to test the bot/subtask after making any package version modifications.

Q: How can I change the output format of the bot?

A: When calling a subtask, the return value is always of type dictionary, and the value(s) returned from the subtask can be access through their specific variable names as a key within the dictionary. Once that value has been returned, the resulting number could be converted to a string should you need it to be.

For questions relating to Enterprise A2019: See the Enterprise A2019 FAQs.

Appendix A: Record of Changes

No.	Version Number	Date of Change	Author	Notes
1	2.0.0.0	06/08/2022	Sikha P	

Appendix B: References

No.	Topic	Reference Link
1	Overview of Enterprise A2019	Click here
2	Guidance: Building basic A2019 bots	Click here
3	Guidance: Building A2019 action packages	Click here
4	APeople Community Forum	Click here
5	Automation Anywhere University	Click here

of