Automation 360 – Upload Files to Azure Portal Bot

Readme

Version 1.0

4/28/2021

Table of Contents

1. Introduction

This document contains all essential information for the user to make full use of this Upload Files to Azure Portal bot . It includes the description of the functions and capabilities and step-by-step procedure for setup & configuration of the bot.

1.1 Overview

Upload Files to Azure Portal Bot can be used to upload files from your local system to Azure Portal.

1.2 Use cases

The key use cases include:

- File Management in Azure Portal

2. Requirements & Prerequisites

2.1 System Requirements

Below are the minimum system requirements for running bots as an Enterprise Automation 360 (Cloud deployed) 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

Automation 360 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 in your machine.
- .Net Framework 4.8 installed

3. Getting Started

3.1 Quick Start

3.1.1 Setup

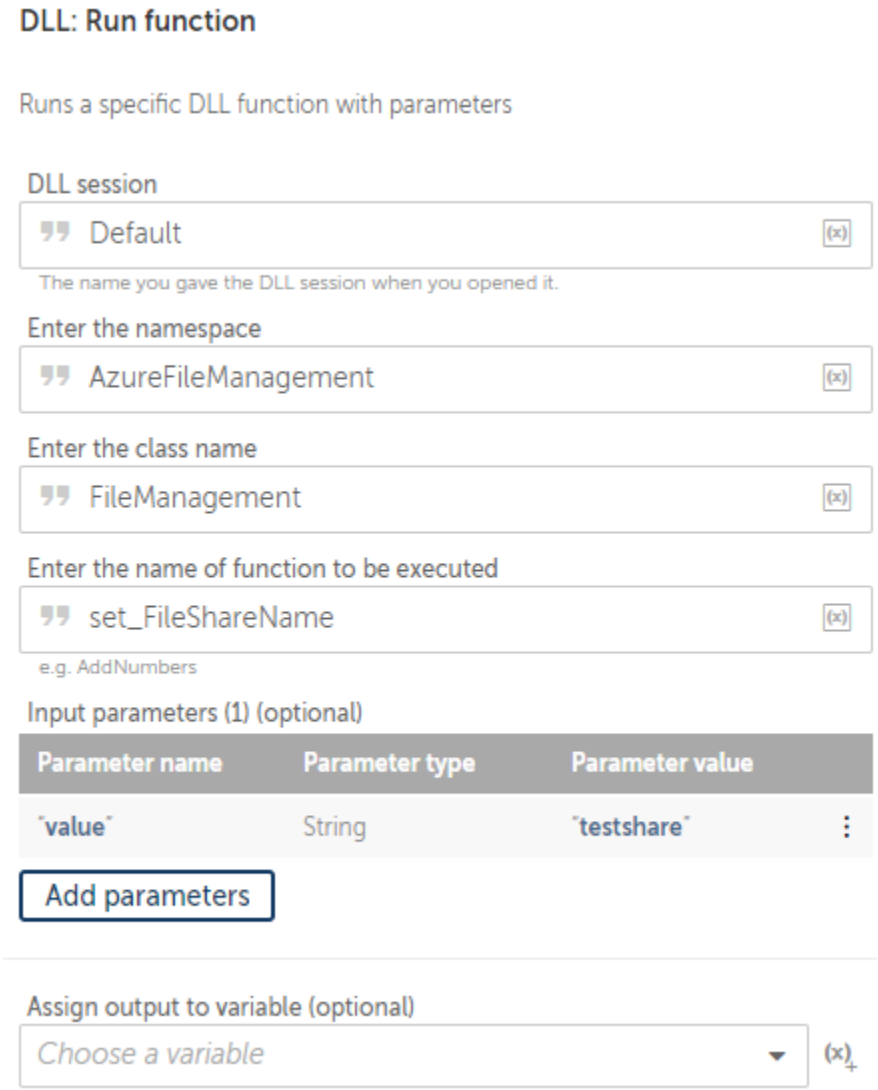
- Install the Bot from Bot Store into your Control Room
- Navigate to Azure File Upload – Automation Anywhere in the Bot Store folder to examine the installed bot.

3.1.2 Configuration and Use

Create a credential locker with the name AzureLocker and create credentials with the AzureCredential. Add Attributes with the names StorageAccountName and StorageAccountKey.

S/N	Variable Name	Variable Type	Function
1	StorageAccountKey	Credential	This is the key that provides access to Azure storage and needs to be stored in Credential Vault.
2	StorageAccountName	Credential	This is the name of your storage account in Azure portal.
3	FileShareName	String	This is the name of your file share in Azure portal. If you mention a file share that does not exist in your Azure portal, it will create one. Here are the guidelines for file share naming conventions in Azure. See this link: https://docs.microsoft.com/en-us/rest/api/storageservices/naming-and-referencing-shares--directories--files--and-metadata
4	FileToUpload	String	This is the name of the file you wish to upload. Example – abc.pdf
5	FilePath	String	This is the local system file path from which you would like to upload the file to Azure portal. Example – “C:/users/user/test.pdf”
6	FolderNameInAzure	String	This is an optional field. If you specify this field, it will upload the file in this folder. If the folder does not exist, it will first create the folder and then upload the file. Example - test

Configure StorageAccountName and StorageAccountKey to get the values from the Credential vault. Add the values for other 4 variables as shown in the screenshots below for your reference.



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

AzureFileManagement

Enter the class name

FileManagement

Enter the name of function to be executed

set_FileToUpload

e.g. AddNumbers

Input parameters (1) (optional)

Parameter name	Parameter type	Parameter value
"value"	String	"ExtractedData.xlsx"

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

AzureFileManagement

Enter the class name

FileManagement

Enter the name of function to be executed

set_FilePath

e.g. AddNumbers

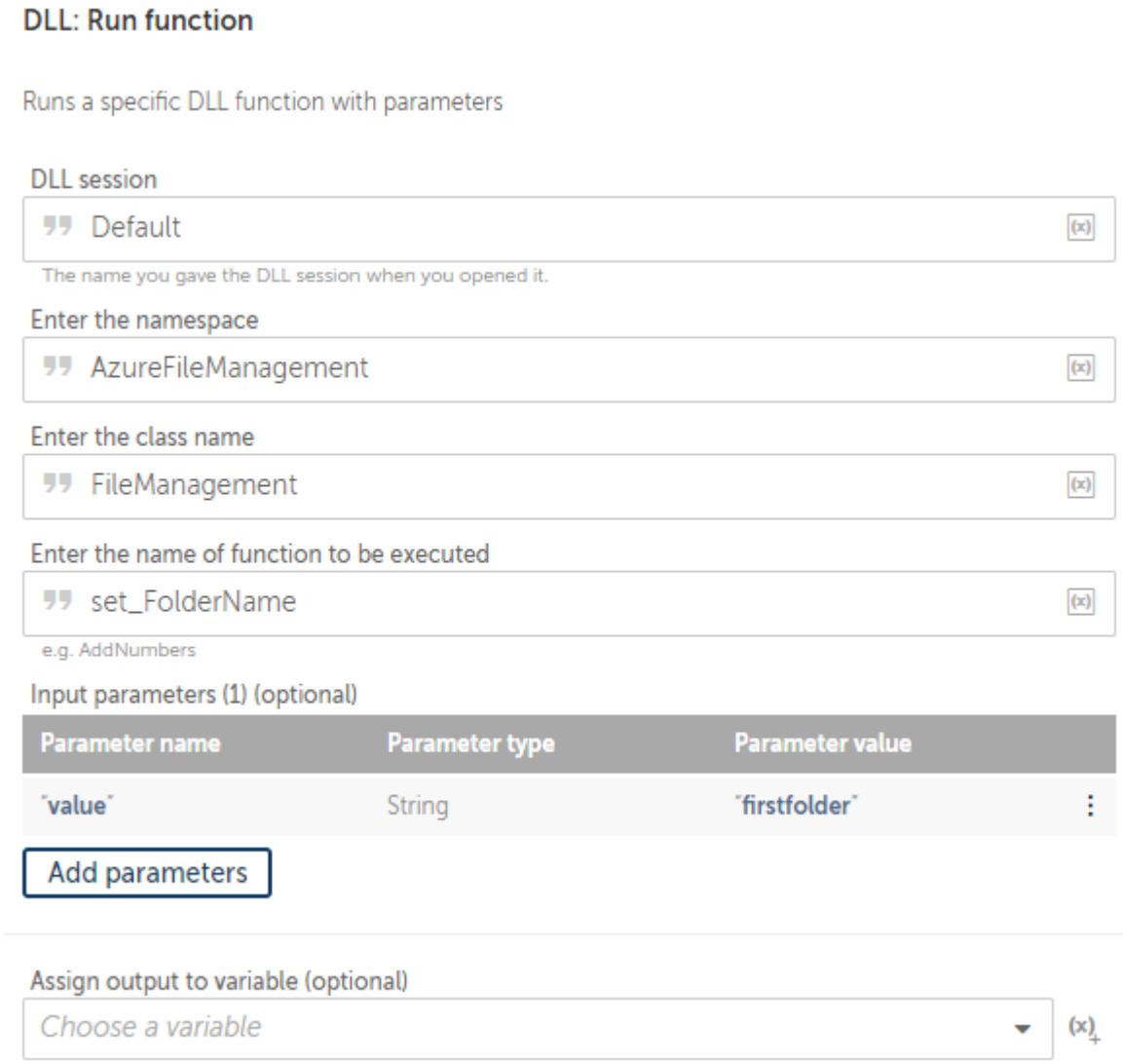
Input parameters (1) (optional)

Parameter name	Parameter type	Parameter value
"value"	String	"C:\temp\ExtractedData.xlsx"

Add parameters

Assign output to variable (optional)

Choose a variable



Run the bot and validate the files in the Azure portal.

4. Support & FAQs

4.1 Support

Free bots and packages 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 Automation 360.

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.

For questions relating to Automation 360: See the Automation 360 FAQs.

Appendix A: Record of Changes

No.	Version Number	Date of Change	Author	Notes
1	1.0.0.0	4/28/2021	Arjun S Meda	Initial version of Bot

Appendix B: References

No.	Topic	Reference Link
1	Overview of Automation 360	Click here
2	Guidance: Building basic Automation 360 bots	Click here
3	Guidance: Building Automation 360 action packages	Click here
4	APeople Community Forum	Click here
5	Automation Anywhere University	Click here