


## Q1. BOT to demonstrate FTP/SFTP

### 1. FTP/SFTP: Connect

- **Server Name:** ftp.dlpctest.com
- **Port:** 21 (default FTP port)
- **Server Type:** FTP
- **Authentication Type:** dlpuser
  - **Username:** dlpuser
  - **Password:** rNrKYTX9g7z3RgJRmxWuGHbeu
- **Transfer Mode:** Passive
- **Transfer Type:** Binary

**Ftp/Sftp: Connect** 

☒ Binary

☐ ASCII

Default directory (optional)

> (x)

Remote folder path

☐ Reconnect if connection fails

Attempts

Time between attempts

---

Create FTP/SFTP session

**Local session** Global session Variable

Local session name can only be used in this bot.

> (x)

### 2. FTP/SFTP: Put Folders

- **Session Name:** FtpSession

- **Local Folder:** C:/Users/user/Downloads/A55
- **File Types (optional):** \*.txt

**Ftp/Sftp: Put folders**

Uploads a folder from a client machine to the FTP/SFTP server  
Required bot agent version: 20.11 or above

FTP/SFTP session

Session name Variable

FtpSession (x)

The name you gave the session when you opened it.

Local folder

C:/Users/user/Downloads/A55 (x)

e.g D:/folder2 , F:/folder1.E:/folder2

File types (optional)

\*.txt (x)

e.g \*.doc , \*.doc\*.txt

### 3. FTP/SFTP: Get Files

- **Session Name:** FtpSession
- **Remote Files:** C:\Users\user\Downloads\A55
- **Local Folder:** C:/folder1
- **Transfer Type:** Binary

**Ftp/Sftp: Get files**

Downloads files from a remote FTP/SFTP folder to a specific location on a client machine  
Required bot agent version: 20.11 or above

FTP/SFTP session

Session name Variable

FtpSession (x)

The name you gave the session when you opened it.

Remote files

/ashish.txt (x)

e.g folder1/abc.doc, folder2/file.txt

Local folder

C:/A55 (x)

e.g C:/folder1

Transfer type

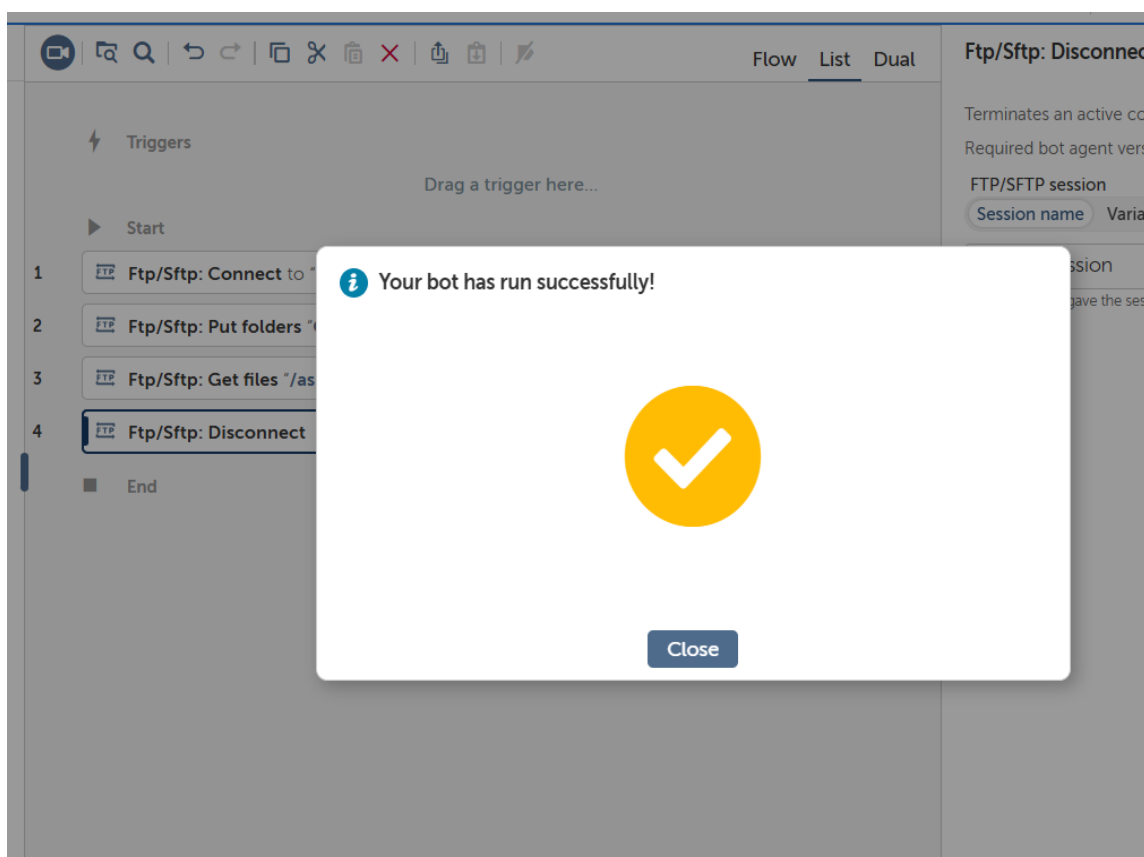
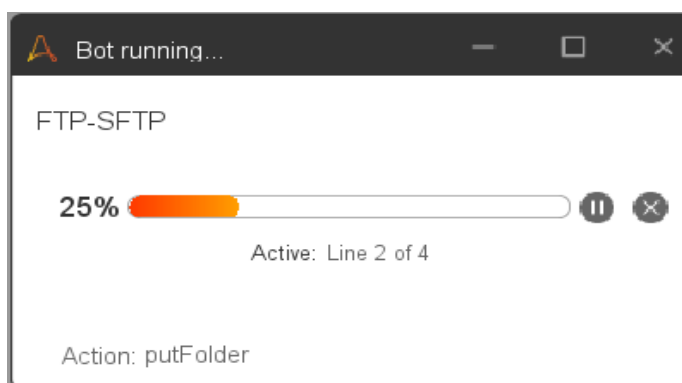
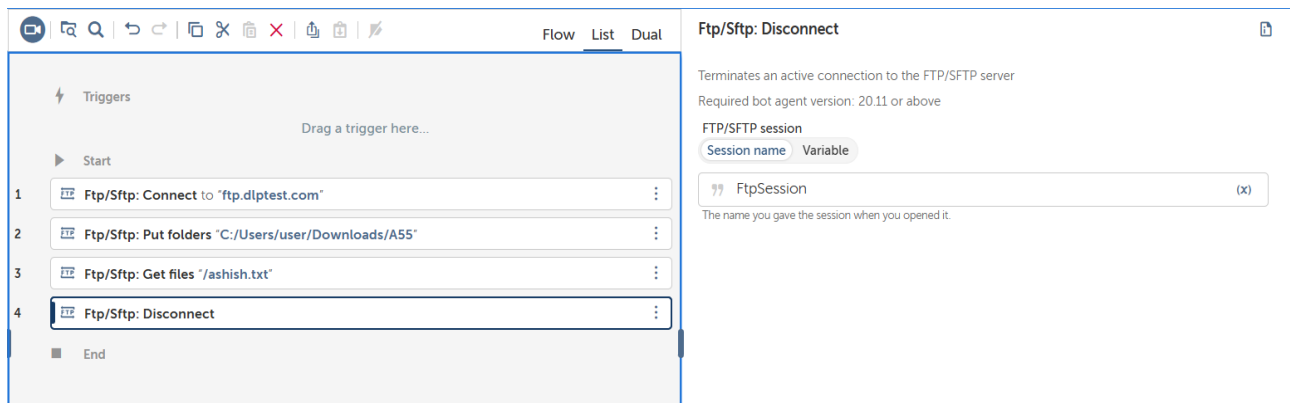
Binary

ASCII

--Select date--

### 4. Disconnect from FTP Server

1. Drag the FTP Disconnect command at the end to close the connection after all operations are complete.



**Step-by-Step: How to check uploaded files on FTP server****1. Use an FTP client software**

Example free tools:

- **FileZilla** (very popular) — [Download FileZilla](#)

**2. Connect to FTP Server**

Open FileZilla → then fill in these details:

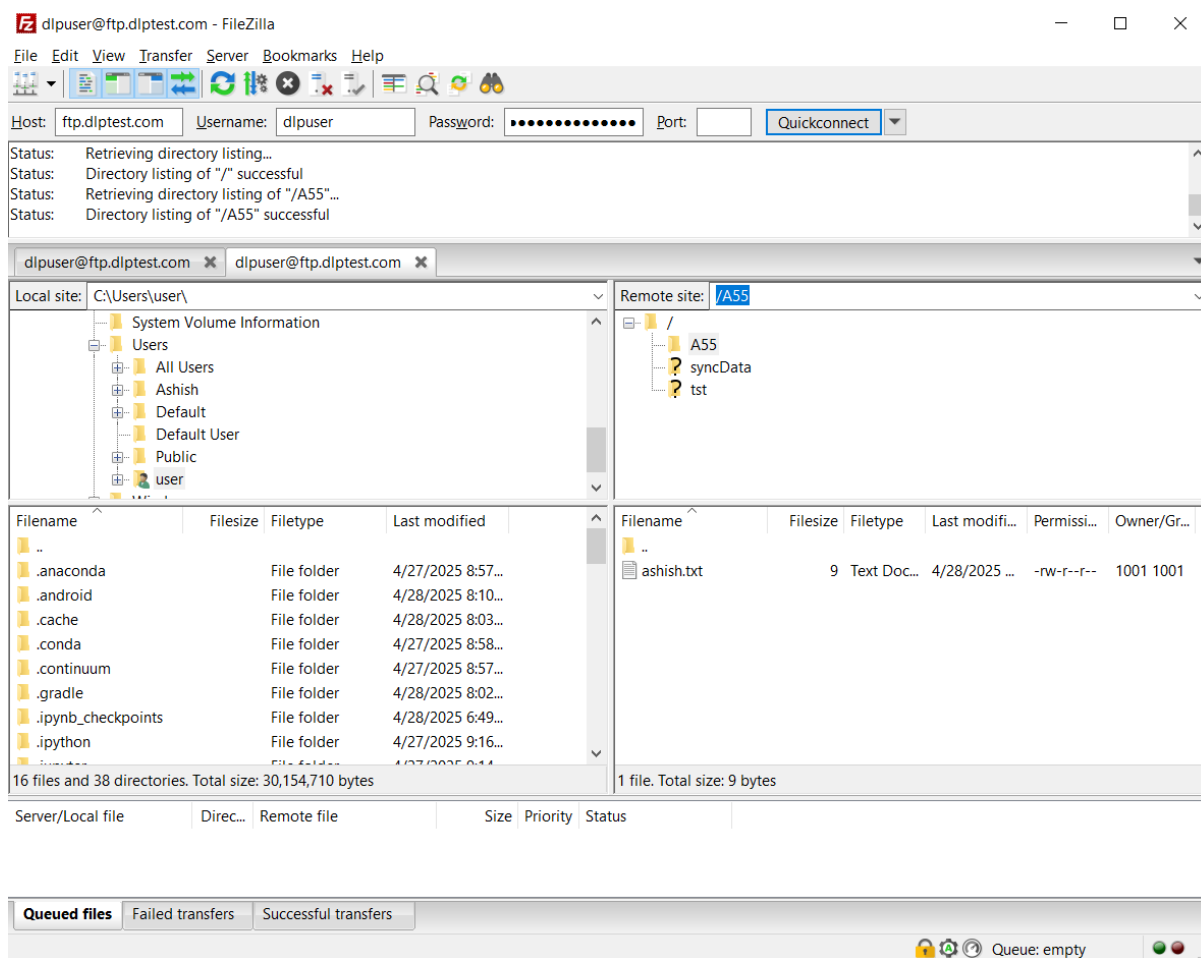
Field	Value
Host	ftp.dlptest.com
Username	dlpuser
Password	rNrKYTX9g7z3RgJRmxWuG Hbeu
Port	21

- Then click Connect.
- Ignore security warning if it comes (because it's a public test server).

**3. Navigate Folders**

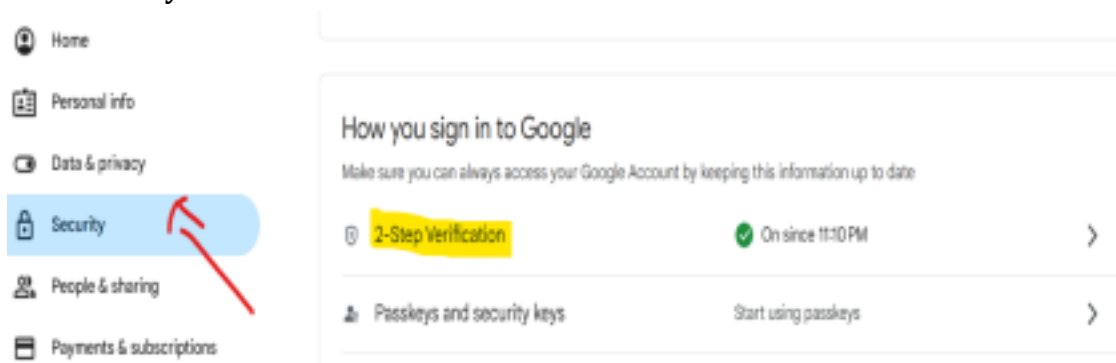
- After connection, you will see server folders on the right side.
- See if your files are uploaded in:
  - / root folder, or
  - /upload folder (some FTP servers default to this)

**4. Look for your .txt files**



## Q2. BOT to send email notification

**Step1: Go to the google account -> Security -> Two step verification -> Add mobile number & verify.**



**Step 2: Open this link <https://support.google.com/mail/answer/185833?hl=en> and give name for that app password**

To create a new app-specific password, type a name for it below...

App name  
**RPA**

Create

## Generated app password

Your app password for your device

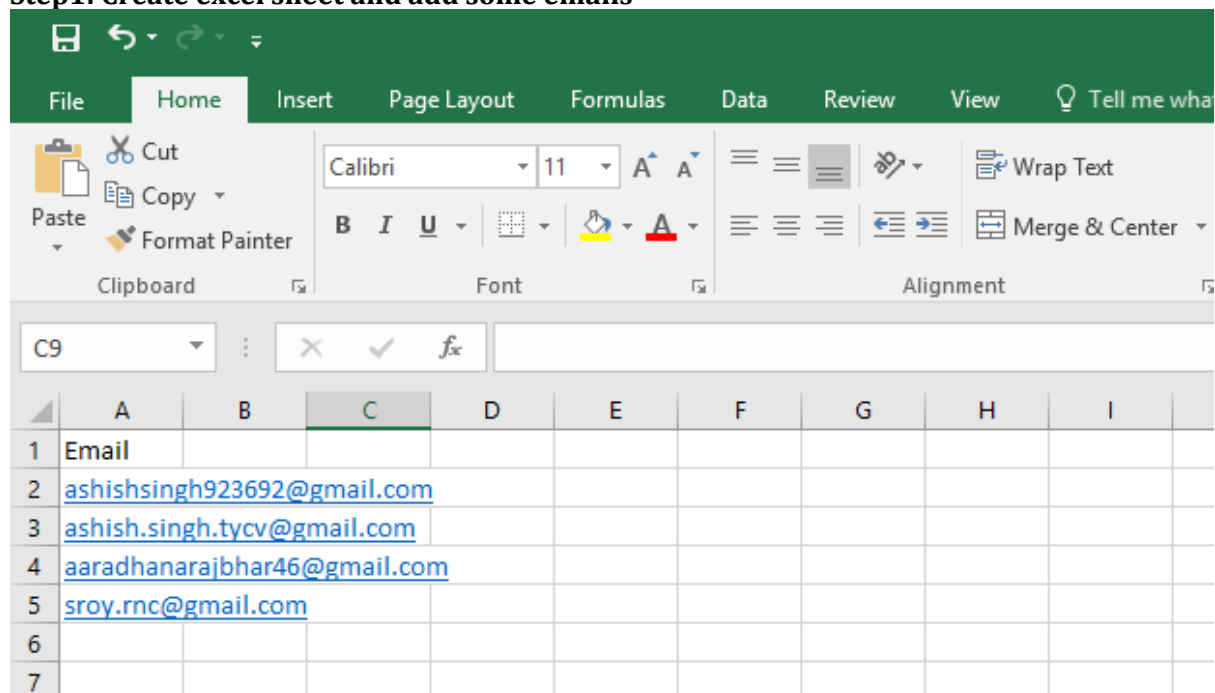
[Redacted password]

### How to use it

Go to the settings for your Google Account in the application or device you are

After that copy the password and save it in notepad.

**Step1: Create excel sheet and add some emails**



The screenshot shows the Microsoft Excel interface. The ribbon is set to 'Home'. The 'Clipboard' group contains icons for Cut, Copy, Paste, and Format Painter. The 'Font' group shows 'Calibri' font and size '11'. The 'Alignment' group includes options for text alignment and 'Merge & Center'. The spreadsheet has columns A through I and rows 1 through 7. The data is as follows:

	A	B	C	D	E	F	G	H	I
1	Email								
2			<a href="mailto:ashishsingh923692@gmail.com">ashishsingh923692@gmail.com</a>						
3			<a href="mailto:ashish.singh.tycv@gmail.com">ashish.singh.tycv@gmail.com</a>						
4			<a href="mailto:aaradhanarajbhar46@gmail.com">aaradhanarajbhar46@gmail.com</a>						
5			<a href="mailto:sroy.rnc@gmail.com">sroy.rnc@gmail.com</a>						
6									
7									

**Step 2: Insert Advance excel package and upload this excel file.**

Triggers

Drag a trigger here...

Start

- Excel advanced: Open "D:\Ashish\email.xlsx"
- Loop : For each row in worksheet and assign to `$ExcelRow$`
- # Number: Increment `$N$` by 1 and assign result to a `$N$` variable
- Excel advanced: Read row From specific cell "ASN.Number.toString\$" and store values to `$ExcelRowData$`
- Email: Send an email to `$ExcelRowData.LegacyAutomation:listToString$` with subject : "Automation Anyw..."
- Message box "Email has send to : `$ExcelRowData.LegacyAutomation:listToString$`"

End

### Step 3: Insert Loop and select each row in worksheet

#### Loop



Runs a sequence of actions for a specific number of times or until a condition is met.

Required bot agent version: 20.11 or above

Label (optional)

&

Loop Type

☒ Iterator

Iterator

Excel advanced

For each row in worksheet

Iterator for each row in Excel. Output(s) will be assigned to a record variable

Loop through

All rows

Read option

☐ Read visible text in cell  
| e.g. 50% will be read as 50%

☒ Read cell value  
| e.g. 50% will be read as 50

Session name

Session name Variable

Excel\_session

(x)

The name you gave the session when you opened it.

Assign the current value to this variable

Multiple variables Record

ExcelRow

9

(x)

**Step 4: Create an Int variable and add default value as 1.**

Edit variable

Cancel

Apply

Type

# Number

Name

N

Maximum = 50 characters

Description (optional)

Digits

Maximum = 255 characters

☐ Use as input☐ Use as output☐ Constant (read-only)

Default value (optional)

1

+

-

**Number: Increment**

Increments (increases by set intervals) a number by a user-specified value

Required bot agent version: 20.18 or above

Enter number

# \$N\$

(x)

Enter increment value

# 1

(x)

Increments number by value (e.g. 1)

Assign the output to variable

# N




(x) +

**Step 5: Insert Increment package**

Start

- Excel advanced: Open "D:\Ashish\email.xlsx" ⋮
- Loop : For each row in worksheet and assign to \$ExcelRow\$ ⋮
- # Number: Increment \$N\$ by 1 and assign result to a \$N\$ variable ⋮



**Number: Increment** 

Increments (increases by set intervals) a number by a user-specified value  
Required bot agent version: 20.18 or above

Enter number

# \$N\$ (x)


Enter increment value

# 1 (x)

Increments number by value (e.g. 1)


---


Assign the output to variable


# N ▼ (x) 

**Step 6: Insert open Excel Advanced Read row package and create a list variable to store the excel data.**


Drag a trigger here...


 **Start**

 **Excel advanced: Open** "D:\Ashish\email.xlsx" ⋮

 **Loop** : For each row in worksheet and assign to \$ExcelRow\$ ⋮

# **Number: Increment** \$N\$ by 1 and assign result to a \$N\$ variable ⋮

 **Excel advanced: Read row** From specific cell "**A\$N.Number:toString\$**" and store val... ⋮

 **Email: Send** an email to \$ExcelRowData.LegacyAutomation:listToString\$ with subje... ⋮

**Excel advanced: Read row**

Extracts data from a row and stores it in a list variable of string data type.

Required bot agent version: 21.112 or above

**Cell option**

- ☐ From active cell
- ☒ From specific cell

**Cell address**

” A\$N.Number:toString\$

(x)

e.g. A5 or B10

- ☐ Read full row

**Read option**

- ☐ Read visible text in cell  
| e.g. 50% will be read as 50%
- ☒ Read cell value  
| e.g. 50% will be read as 50

**Session name**

Session name Variable

” Excel\_session

(x)

The name you gave the session when you opened it.

**Assign the output to variable**

ExcelRowData

(x)

**Step 7: Insert Email Send package and add following configuration details**

Start

- 1 Excel advanced: Open "D:\Ashish\email.xlsx"
- 2 Loop : For each row in worksheet and assign to \$ExcelRow\$
- 3 # Number: Increment \$N\$ by 1 and assign result to a \$N\$ variable
- 4 Excel advanced: Read row From specific cell "A\$N.Number:toString\$" and store val...
- 5 Email: Send an email to \$ExcelRowData.LegacyAutomation:listToString\$ with subje...

**Email: Send**

Sends an email to one or more recipients from Microsoft Outlook or a mail server. Attach files and include email body in plain text, HTML code or rich text format.

Required bot agent version: 21.210 or above

**To address**

” \$ExcelRowData.LegacyAutomation:listToString\$ (x)

Use comma for multiple email ids

**Cc (optional)**

” (x)

Use comma for multiple email ids

**Bcc (optional)**

” (x)

Use comma for multiple email ids

☐ Raise error if the email ids are invalid (Email Server only)

**Subject**

” Automation Anywhere finished executing the TaskBot (x)

**Attachment (optional)**

List Variable

(x)

☐ Raise error if the attachments are missing(Desktop only)

**Email body**

Plain text HTML design HTML code

**Message**

” Email Test (x)

**Email: Send**☒ Include Go Green message at the end of the email

Send email via

**Email server** Outlook EWS Server

Encoding for email (optional)

UTF-8

From address

” ashish73737447@gmail.com

(x)

Use secure connection (SSL/TLS)

False **True** Variable

Email server host

” smtp.gmail.com

(x)

eg: smtp-mail.outlook.com,smtp.gmail.com ,etc.

Email server port

# 587

(x)

eg: 587

My server requires authentication

False **True** Variable

Authentication mode

Basic

Username (optional)

Credential Variable **Insecure string**

” ashish73737447@gmail.com

(x)

Password (optional)

Credential Variable **Insecure string**

” ikfbxfvdurhvjvo

(x)

**Step 8: Add Message box to display successfully email has been send message.****Message box**

Inserts a message box to show a message when the task runs

Required bot agent version: 20.11 or above

Enter the message box window title

” Automation Anywhere Enterprise Client

(x)

Enter the message to display

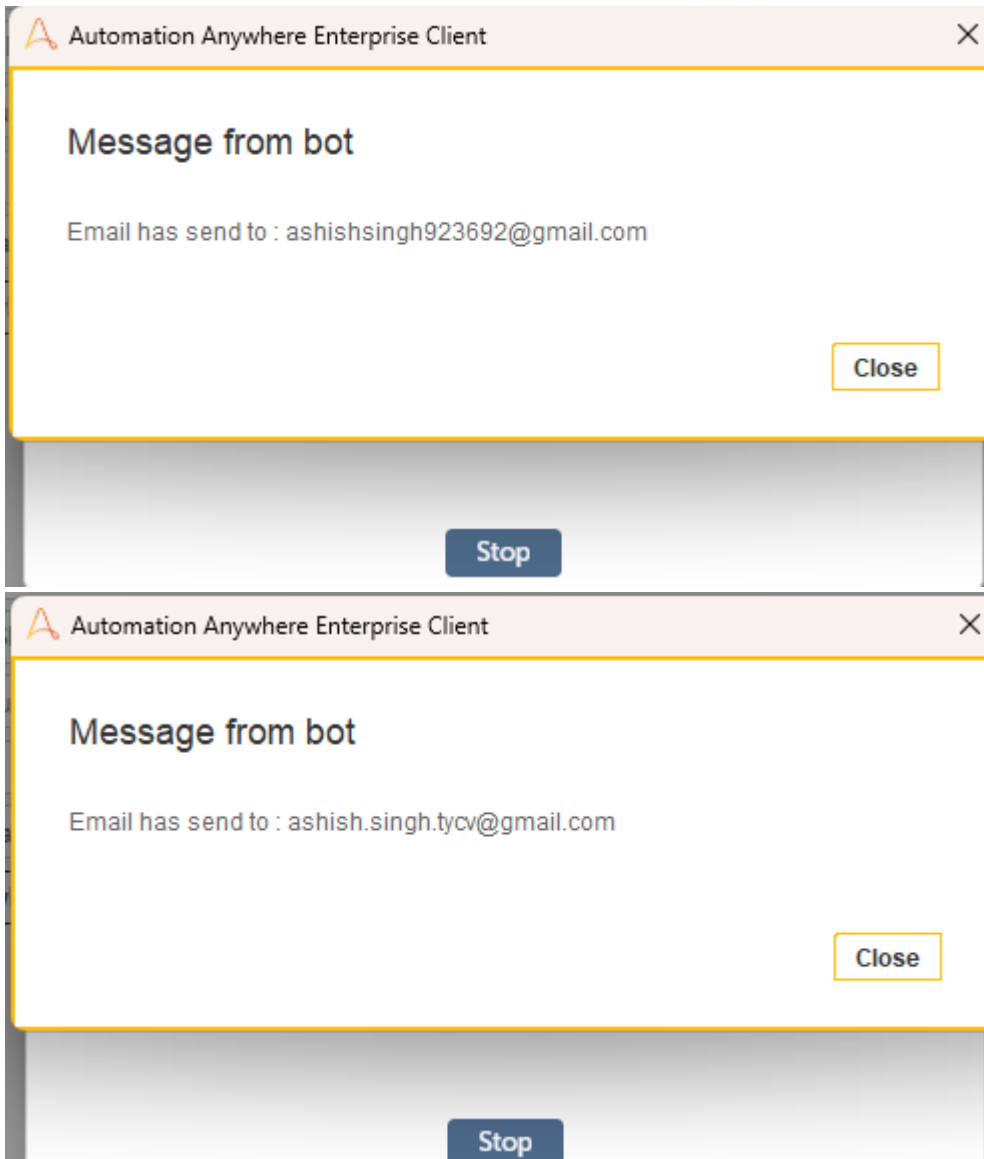
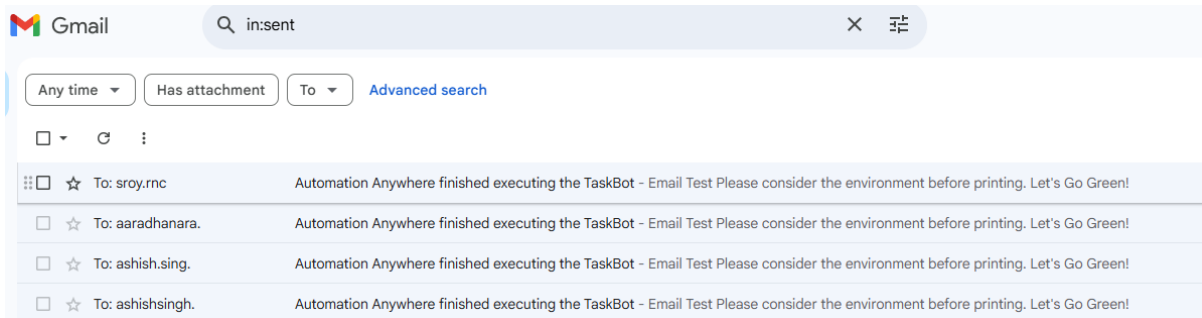
” Email has send to : \$ExcelRowData.LegacyAutor

(x)

Scrollbar after lines



# 30

(x)



**Q3. BOT to demonstrate PGP(Pretty Good Privacy) to encrypt and decrypt file contents.****Actions**

## ✓ PGP

-  Create keys
-  Decrypt files
-  Encrypt files

Create Keys action helps us to create public and private keys.

Decrypt Files action is used to decrypt the already encrypted files.

Encrypt Files action encrypts the file using one of the specified Encryption Algorithm.

**Step 1: Select Action PGP: Create keys**

- a) Provide Location to save Public key file C:\Users\Admin\Desktop\Ashish\Public\_Key.txt
- b) Provide Location to save Private key file C:\Users\Admin\Desktop\Ashish\Private\_Key.txt
- c) Set the Password to protect private key file (optional) as 'Insecure String'

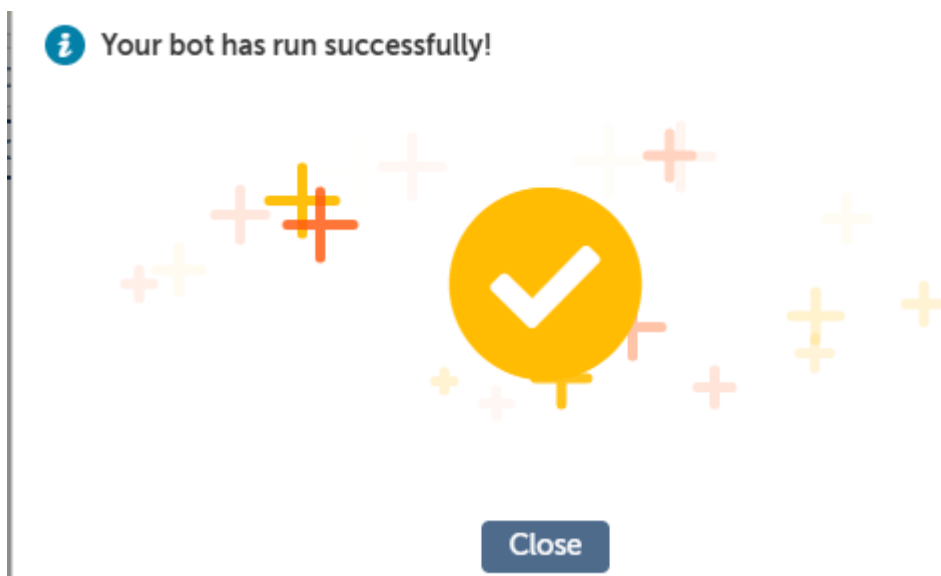
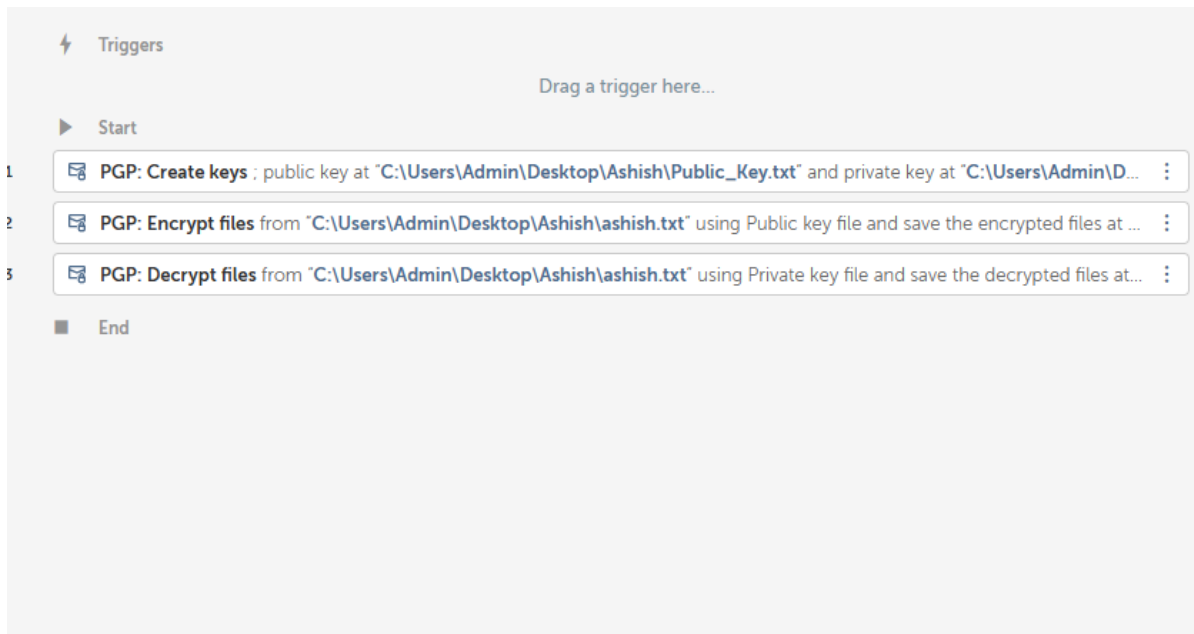
**Step 2: Select Action PGP: Encrypt files**

- a) Provide Encryption Type as Public Key File  
C:\Users\Admin\Desktop\Ashish\Public\_Key.txt
- b) Provide Encryption algorithm as AES256
- c) Provide the source file/folder C:\Users\Admin\Desktop\Ashish\ashish.txt
- d) Provide the destination file/folder C:\Users\Admin\Desktop\Ashish\ashish12.txt

**Step 3: Select Action PGP: Decrypt files**

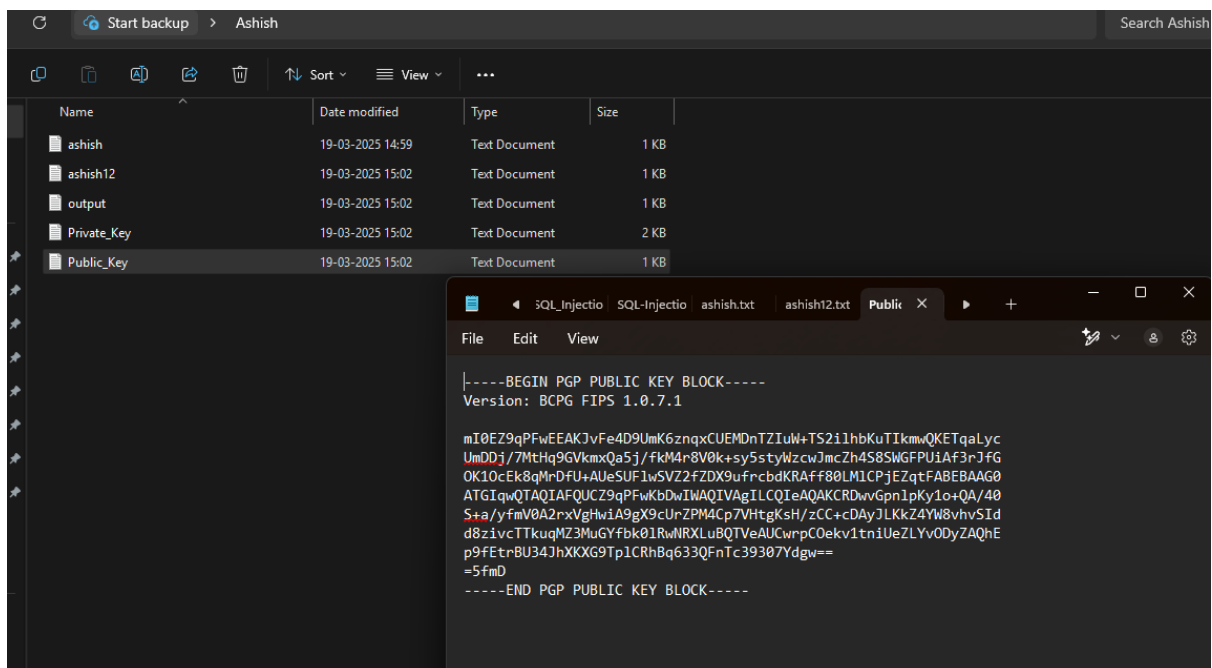
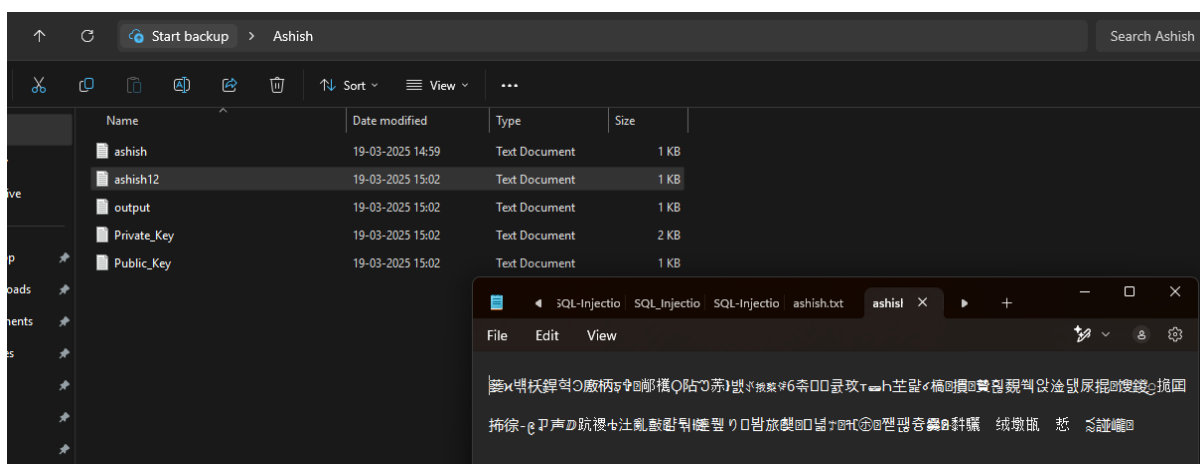
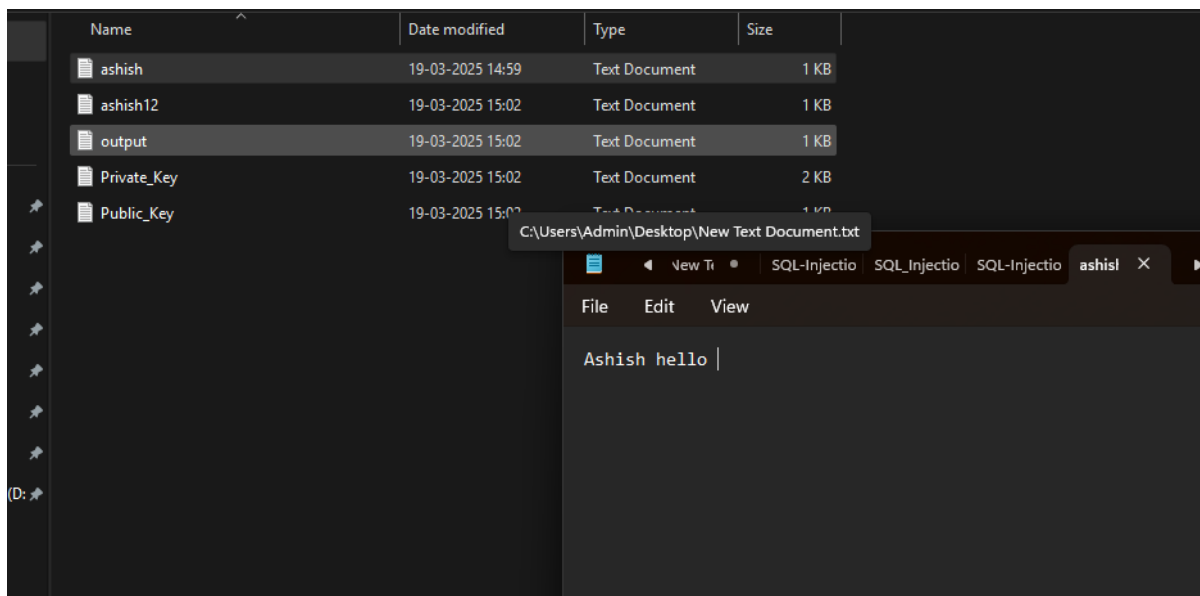
- a) Provide Encryption Type as Private Key File  
C:\Users\Admin\Desktop\Ashish\Private\_Key.txt
- b) Set Password to protect private key file (optional) as Insecure String
- c) Provide the source file/folder C:\Users\Admin\Desktop\Ashish\ashish12.txt
- d) Provide the destination file/folder C:\Users\Admin\Desktop\Ashish\output.txt

**Bot Name: PGP\_Secure\_Files**

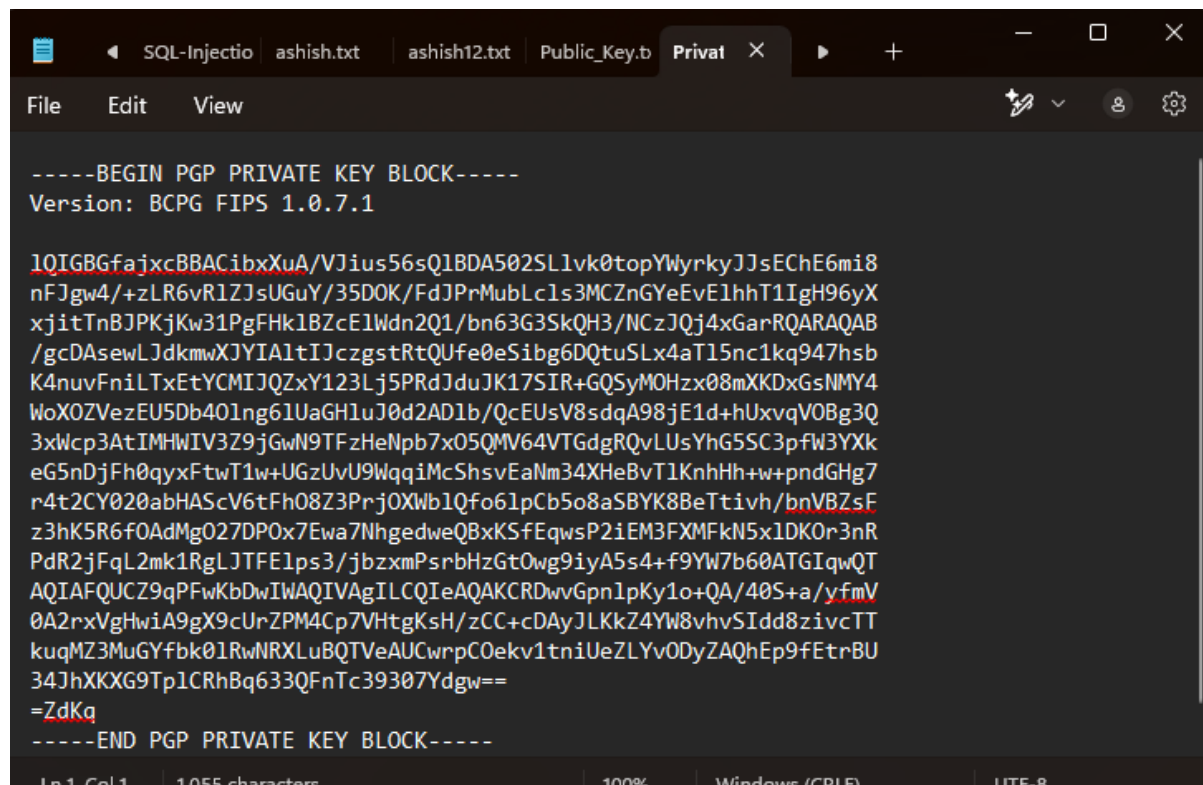


The screenshot shows a file explorer window with the address bar set to 'Start backup > Ashish'. The window displays a list of files and folders:

Name	Date modified	Type	Size
ashish	19-03-2025 14:59	Text Document	1 KB
ashish12	19-03-2025 15:02	Text Document	1 KB
output	19-03-2025 15:02	Text Document	1 KB
Private_Key	19-03-2025 15:02	Text Document	2 KB
Public_Key	19-03-2025 15:02	Text Document	1 KB



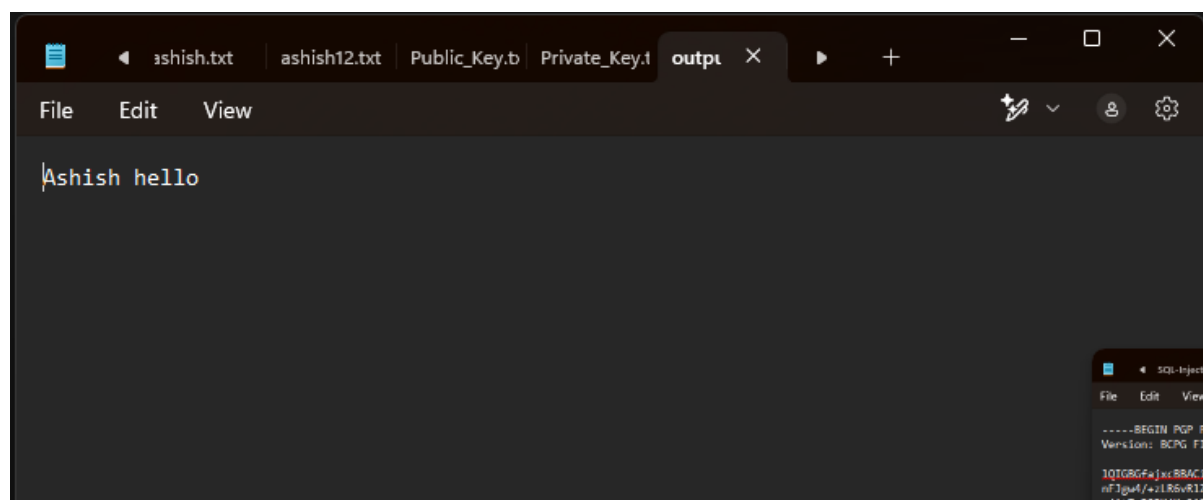




A screenshot of a text editor window with a dark theme. The title bar shows several tabs: 'SQL-Injection', 'ashish.txt', 'ashish12.txt', 'Public\_Key.b', and 'Privat'. The 'Privat' tab is active. The menu bar includes 'File', 'Edit', and 'View'. The text content is a PGP private key block, starting with '-----BEGIN PGP PRIVATE KEY BLOCK-----' and 'Version: BCPG FIPS 1.0.7.1'. The key material is a long string of base64-encoded text. The block ends with '-----END PGP PRIVATE KEY BLOCK-----'. The status bar at the bottom indicates 'Ln 1, Col 1', '1,055 characters', '100%', 'Windows (CRLF)', and 'UTF-8'.

```
-----BEGIN PGP PRIVATE KEY BLOCK-----
Version: BCPG FIPS 1.0.7.1

lQIGBGfajxcBBACibxXuA/VJius56sQlBDA502SLlvk0topYWyrkyJJSEChE6mi8
nFJgw4/+zLR6vR1ZJ5UGuY/35DOK/FdJPrMubLc1s3MCZnGYeEvE1hhT1Igh96yX
xjitTnBJPKjKw31PgFhk1BZcElWdn2Q1/bn63G3SkQH3/NCzJQj4xGarRQARAQAB
/gcDAsewLJdkmwXJYIAItIJczgstRtQUfe0eSibg6DQtuSLx4aT15nc1kq947hsb
K4nuvFniLTxEtYCMIJQZxY123Lj5PRdJduJK17SIR+GQSyMOHzx08mXKDxGsNMY4
WoX0ZVezEU5Db401ng61UaGHluJ0d2AD1b/QcEUsv8sdqA98jE1d+hUxvqV0Bg3Q
3xWcp3AtIMHWIV3Z9jGwN9TFzHeNpb7x05QMV64VTGdgrQvLUYhG5SC3pfW3YXk
eG5nDjFh0qyxftwT1w+UGzUvU9WqqiMcShsvEaNm34XHeBvTlKnhHh+w+pndGHg7
r4t2CY020abHAScV6tFh08Z3PrjOXWblQfo61pCb5o8aSBYK8BeTtivh/bnVBZsF
z3hK5R6f0AdMg027DPOx7Ewa7NhgedweQBxKSfEqwsP2iEM3FXMFkN5x1DKOr3nR
PdR2jFqL2mk1RgLTJTFE1ps3/jbzxmPsrBHzGt0wg9iyA5s4+f9YW7b60ATGIqwQT
AQIAFQUCZ9qPFwKbDwIWAQIVAgILCQIeAQAKCRDwvGpnlpKy1o+QA/40S+a/yfmV
0A2rxVgHwiA9gX9cUrZPM4Cp7VHtgKsH/zCC+cDAyJLKkZ4YW8vhvSIdd8zivcTT
kuqMZ3MuGYfbk01RwNRXLuBQTVeAUCwrpCOekv1tniUeZLYvODyZAQhEp9fEtrBU
34JhXKXG9Tp1CRhBq633QFnTc39307Ydgw==
=ZdKq
-----END PGP PRIVATE KEY BLOCK-----
```

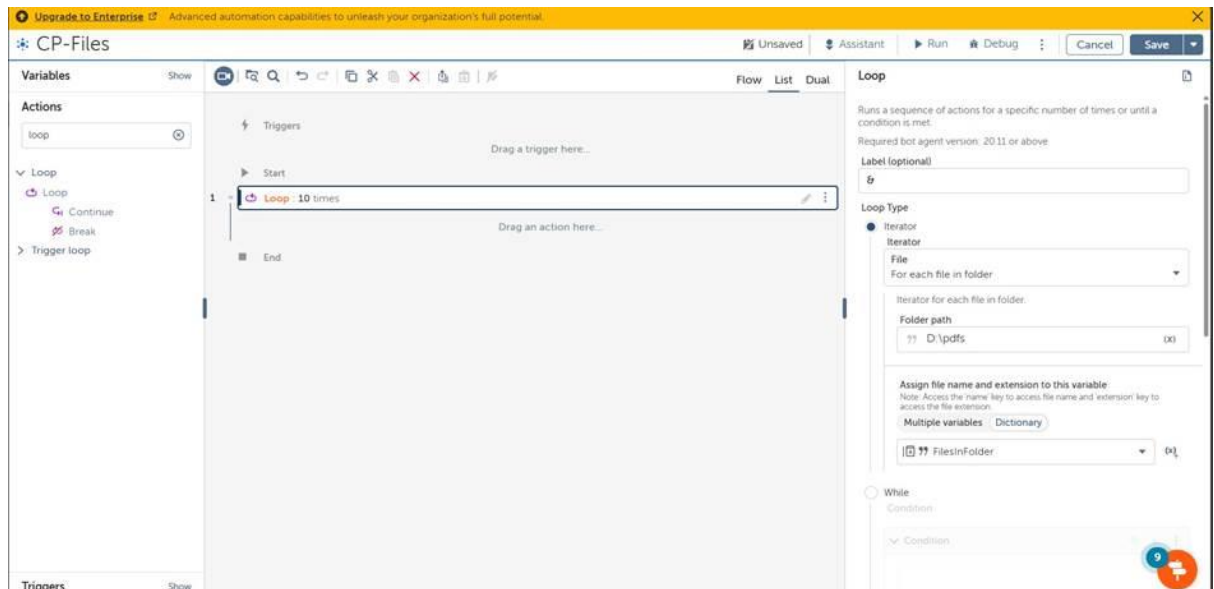


A screenshot of a text editor window with a dark theme. The title bar shows tabs: 'ashish.txt', 'ashish12.txt', 'Public\_Key.b', 'Private\_Key.t', and 'output'. The 'output' tab is active. The menu bar includes 'File', 'Edit', and 'View'. The text content is a single line: 'Ashish hello'. The status bar at the bottom is partially visible, showing 'Ln 1, Col 1', '1,055 characters', '100%', 'Windows (CRLF)', and 'UTF-8'. A small preview of the previous screenshot is visible in the bottom right corner.

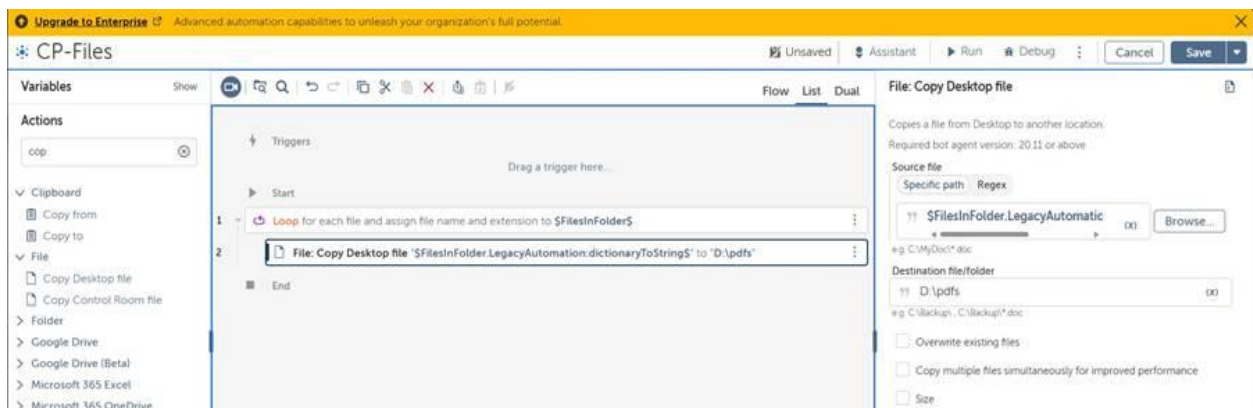
```
Ashish hello
```

#### Q4. Automate task of copying files from a source folder to destination folder.

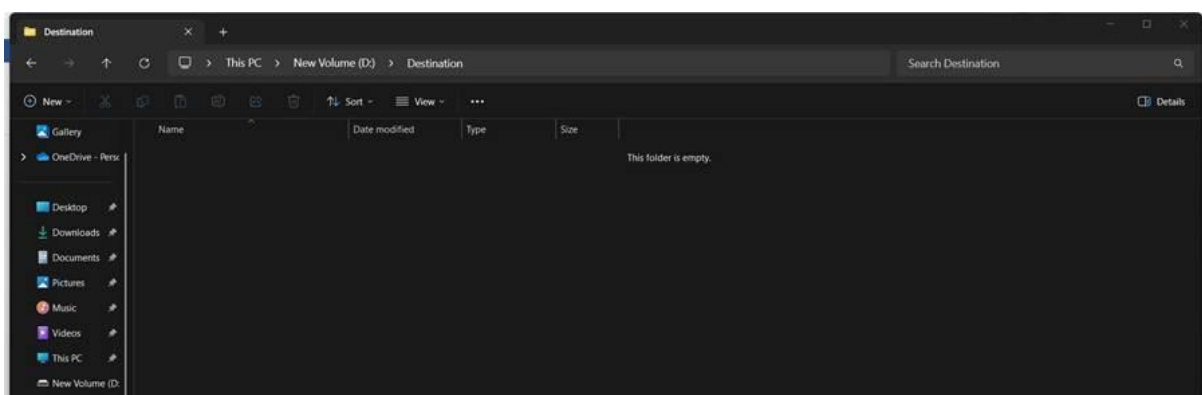
Take loop select each file in folder create a dictionary variable



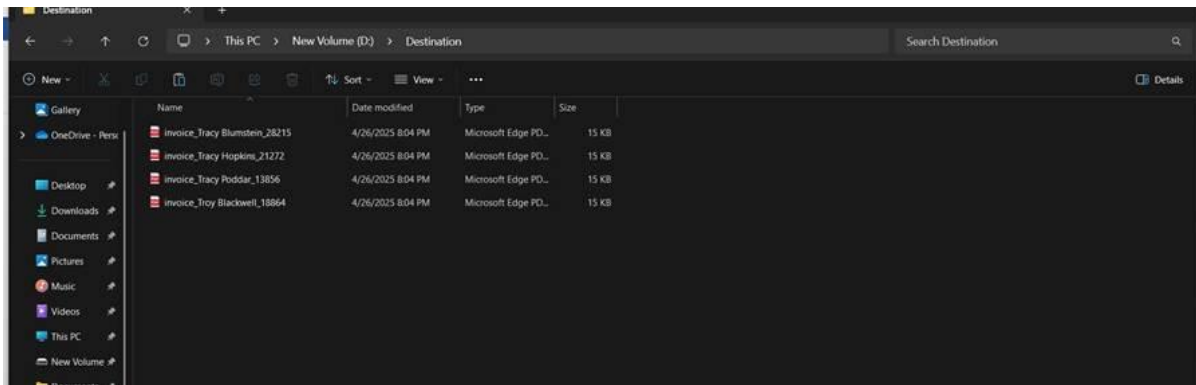
Put file copy action



Before bot



After bot



## Q5. Extract a table from webpage.

### 1. Create a New Bot:

- Open Automation Anywhere Control Room.
- Go to "My Bots" and click on "Create a Bot."

Create Task Bot

Use a Task Bot for desktop and cloud applications. It can be invoked from a Process or another Task Bot.

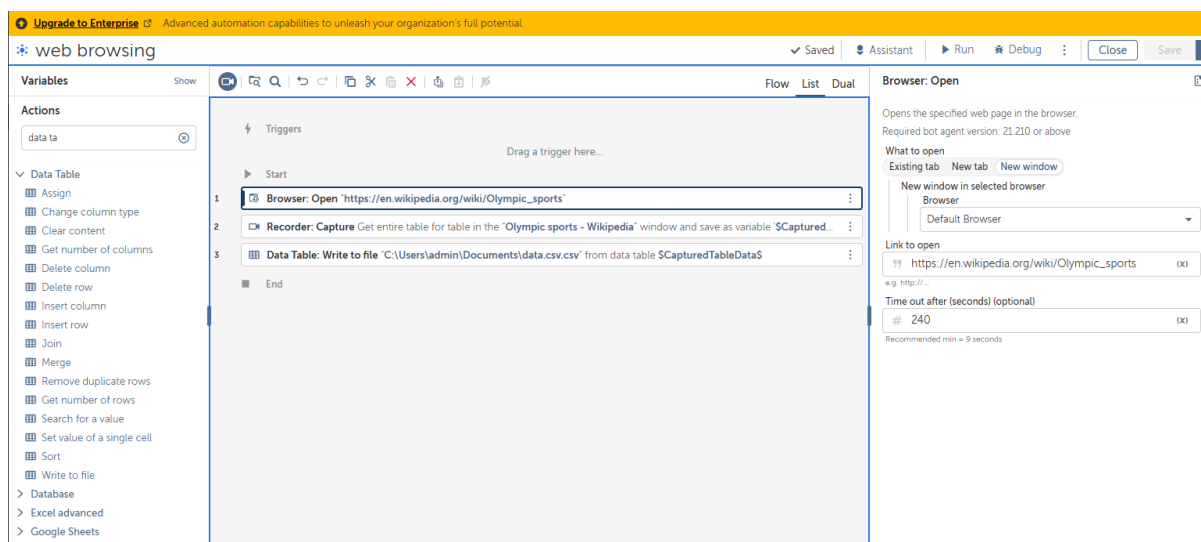
Name:  Description (optional):

Maximum = 50 characters Maximum = 255 characters

Folder:

### 2. Open Browser:

- Drag the **Open Browser** action from the Actions panel.
- Enter the URL of the webpage containing the table you want to extract.



### 3. Capture the Table:

- Search for and drag the **Capture** action (under Web Automation).
- Use the Capture Tool to highlight and select the table from the webpage.
- Choose "Table" when the capture type is prompted.

The screenshot shows the 'web browsing' tool interface. On the left, the 'Actions' panel lists various actions, including 'Data Table' actions like 'Assign', 'Change column type', 'Clear content', 'Get number of columns', 'Delete column', 'Delete row', 'Insert column', 'Insert row', 'Join', 'Merge', 'Remove duplicate rows', 'Get number of rows', 'Search for a value', 'Set value of a single cell', 'Sort', and 'Write to file'. The main workspace shows a workflow with three steps: 1. 'Browser: Open "https://en.wikipedia.org/wiki/Olympic\_sports"', 2. 'Recorder: Capture Get entire table for table in the "Olympic sports - Wikipedia" window and save as variable "\$Captured..."', and 3. 'Data Table: Write to file "C:\Users\admin\Documents\data.csv.csv" from data table "\$CapturedTableData\$"'.

The screenshot shows the Wikipedia page for the 2024 Summer Olympics. The table lists sports and disciplines, with columns for the number of events in each discipline. The table is organized by sport, with disciplines listed in the first column and the number of events in the subsequent columns.

Sport	Discipline	Code & Pictogram	Body	96	00	04	08	12	20	24	28	32	36	48	52	56	60	64	68	72	76	80	
Aquatics	Artistic swimming	SWA	World Aquatics																				
	Diving	DIV				2	1	2	4	5	5	4	4	4	4	4	4	4	4	4	4	4	
	Marathon swimming	OWS																					
	Swimming	SWM				4	7	9	4	6	9	10	11	11	11	11	13	15	18	29	29	26	26
	Water polo	WPO			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Archery		ARC	World Archery			7	6	3	10											2	2	2	
Athletics		ATH	World Athletics			12	23	25	21	26	30	29	27	27	29	29	33	33	34	36	36	37	38
Badminton		BDM	BWF																				
Baseball and softball	Baseball	BSB	WBSC <sup>[s 1]</sup>			*			*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	Softball	SBL																					
Basketball	3x3	BK3	FIBA																				
	Basketball	BKB				*				*		1	1	1	1	1	1	1	1	1	1	2	2
Boxing		BOX	World Boxing <sup>[s 2]</sup>			7	5	8	8	8	8	8	8	8	10	10	10	10	11	11	11	11	
Canoeing	Sprint	CSP	ICF							*		9	9	9	9	7	7	7	7	7	11	11	
	Slalom	CSL																			4		
Cricket		CKT	ICC			1																	

1. \* The World Baseball Softball Confederation, which currently governs both baseball and softball, was created by a 2013 merger of two former governing bodies – the [International Baseball Federation](#) and the [International Softball Federation](#). Baseball and softball were governed

**web browsing** [Saved] [Assistant] [Run] [Debug] [Close] [Save]

**Variables** [Show] [Flow] [List] [Dual]

**Actions**

- data ta
- Data Table
  - Assign
  - Change column type
  - Clear content
  - Get number of columns
  - Delete column
  - Delete row
  - Insert column
  - Insert row
  - Join
  - Merge
  - Remove duplicate rows
  - Get number of rows
  - Search for a value
  - Set value of a single cell
  - Sort
  - Write to file

**Triggers**

Drag a trigger here...

Start

1. Browser: Open 'https://en.wikipedia.org/wiki/Olympic\_sports'
2. Recorder: Capture Get entire table for table in the 'Olympic sports - Wikipedia' window and save as variable '\$Captured...'
3. Data Table: Write to file 'C:\Users\admin\Documents\data.csv.csv' from data table '\$CapturedTableData\$'

End

**Data Table: Write to file**

Writes data from a Table type variable to a CSV or TXT file.  
Required bot agent version: 20.11 or above

Data table name: CapturedTableData (x)

Enter file name: C:\Users\admin\Documents\data.csv.csv (x) [Browse...]  
Required extensions: ".csv", ".txt"

☐ Create folders/files if it doesn't exist

When writing:  
☐ Append to the existing file  
☒ Overwrite existing file

Row delimiter: new line

Column delimiter: comma

Encoding: ANSI

## Output:

data.csv - Excel (Product Activation Failed)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1	hide	Discipline Code & Pi	Body		96	0	4	6	8	12	20	24	28	32	36	48	52	56	60	64	68	72	76	80
2	Aquatics	Artistic sw	SWA	World Aquatics	2	1	2	4	5	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4
3	Diving	DIV																						
4	Marathon	OWS																						
5	Swimming	SWM			4	7	9	4	6	9	10	11	11	11	11	11	13	15	18	29	29	26	26	29
6	Water pol	WPO			1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	Archery	ARC		World Archery		7	6		3		10											2	2	2
8	Athletics	ATH		World Ath	12	23	25	21	26	30	29	27	27	29	29	33	33	33	34	36	36	38	37	38
9	Badminto	BDM		BWF																				
10	Baseball a	Baseball	BSB	WBSC[s 1]																				
11	Softball	SBL																						
12	Basketbal	3x3	BK3	FIBA																				
13	Basketbal	BK8																						
14	Boxing	BOX		World Boxing[s 2]			7		5		8	8	8	8	8	8	10	10	10	10	11	11	11	11
15	Canoeing	Sprint	CSP	ICF																				
16	Slalom	CSL																						
17	Cricket	CKT		ICC		1																		
18	Cycling	BMX freest	BMF	UCI																				
19	BMX racin	BMX																						
20	Mountain	MTB																						
21	Road	CRD																						
22	Track	CTR																						
23	Equestrian	Dressage	EDR	FEI	5	3	7	5	7		4	4	4	4	4	4	4	4	5	5	5	4	4	5
24	Eventing	EVE																						
25	Jumping	EJP				3					2	2	2	2	2	2	2	2	2	2	2	2	2	2
26	Fencing	FEN		FIE	3	7	5	8	4	5	6	7	7	7	7	7	7	7	8	8	8	8	8	8
27	Field hock	HOC		FIH							1		1	1	1	1	1	1	1	1	1	1	1	2
28	Flag footb	AFB		IFAF																				
29	Football	FBL		FIFA			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

## Q6. Bot to extract text from notepad file and copying it into excel.

**Note-cp-excel** [Saved] [Assistant] [Run] [Debug] [Close] [Save]

**Variables** [Show] [Flow] [List] [Dual]

**Actions**

- write
- Data Table
  - Write to file
- Excel advanced
  - Write from data table
- Google Sheets
- Microsoft 365 Excel

**Triggers**

Drag a trigger here...

Start

1. CSV/TXT: Open 'C:\Users\omkar\Downloads\Log.txt.txt'
2. CSV/TXT: Read data and assign to \$TableFromCSV\$
3. Excel advanced: Write from data table

End

**CSV/TXT: Open**

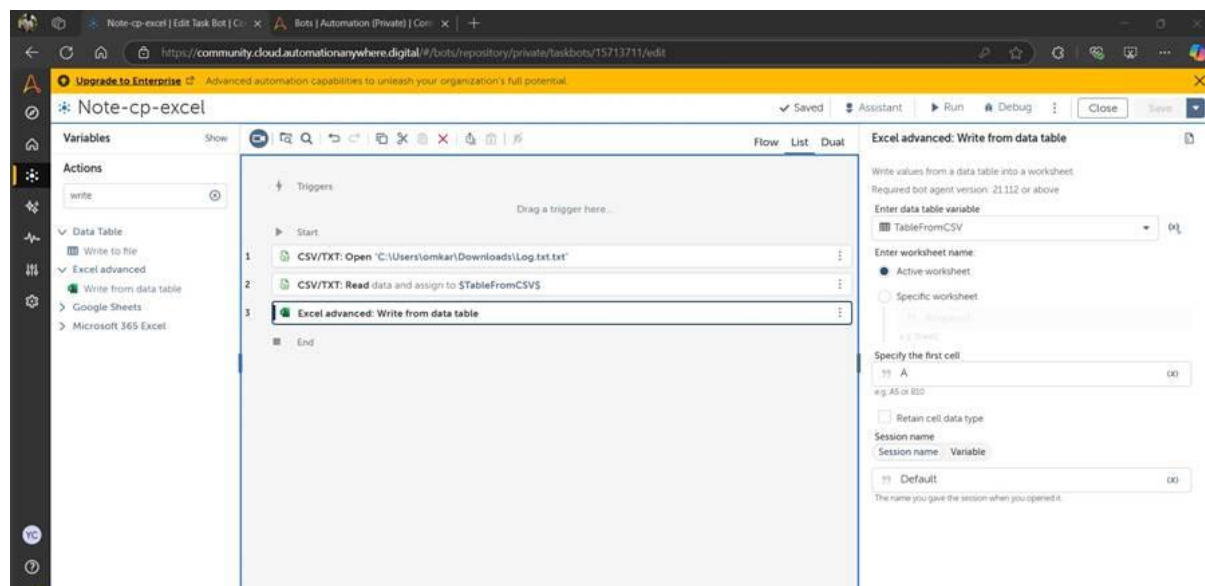
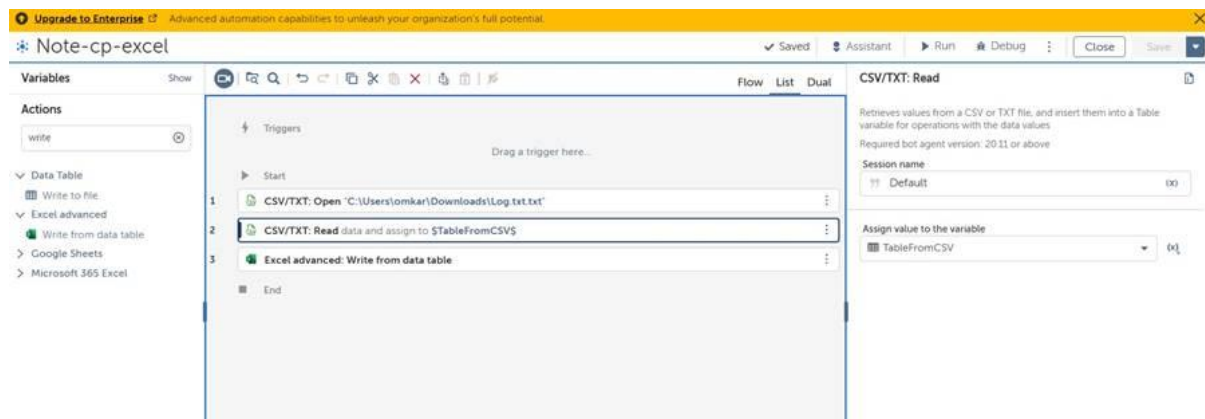
Specifies the delimiter used in the file, whether to trim the spaces, and the encoding applied on the file.  
Required bot agent version: 20.11 or above

Session name: Default (x)

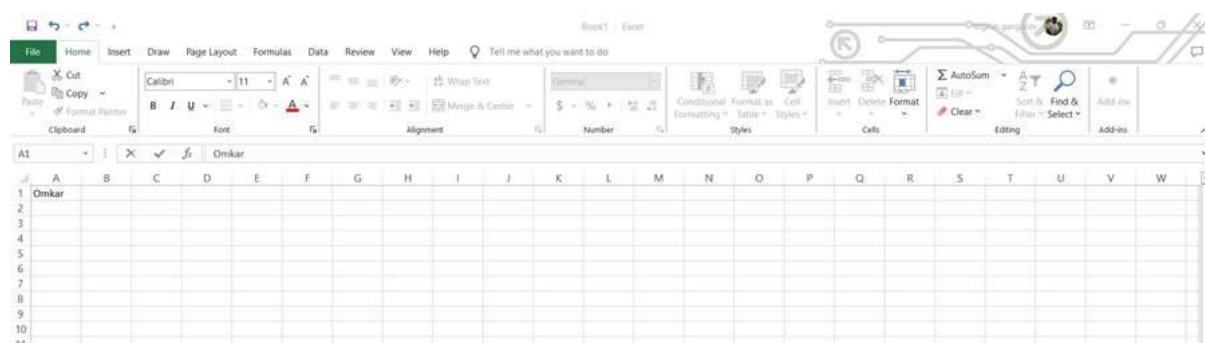
File path: Variable Control Room file Desktop file  
C:\Users\omkar\Downloads\Log (x) [Browse...]  
Required extensions: ".csv", ".txt", ".tsv"

☐ Contains header

Delimiter:  
☒ Comma  
☐ Tab  
☐ Regional list separator  
☐ Newline  
☐ Other [Specify Delimiter optional]

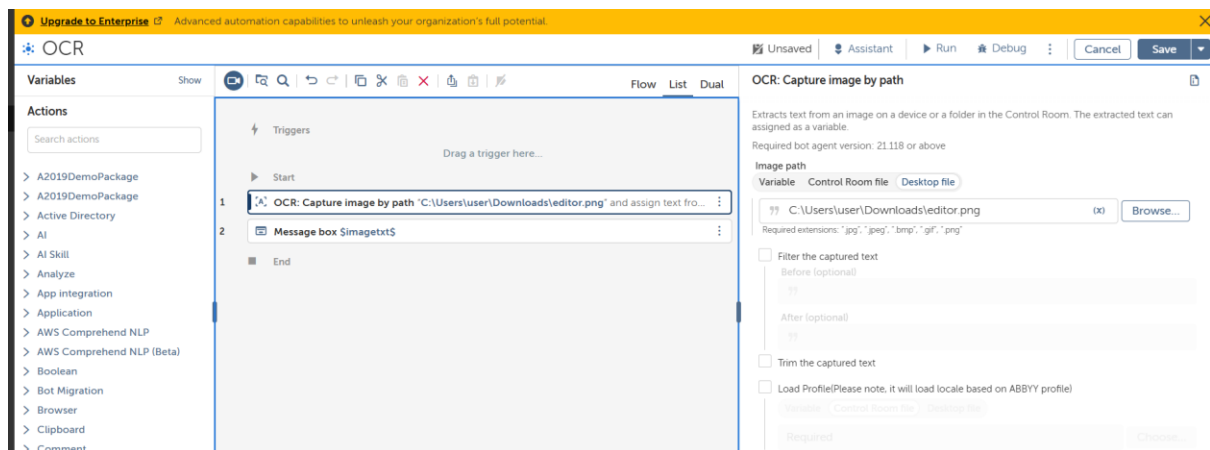


## Output:

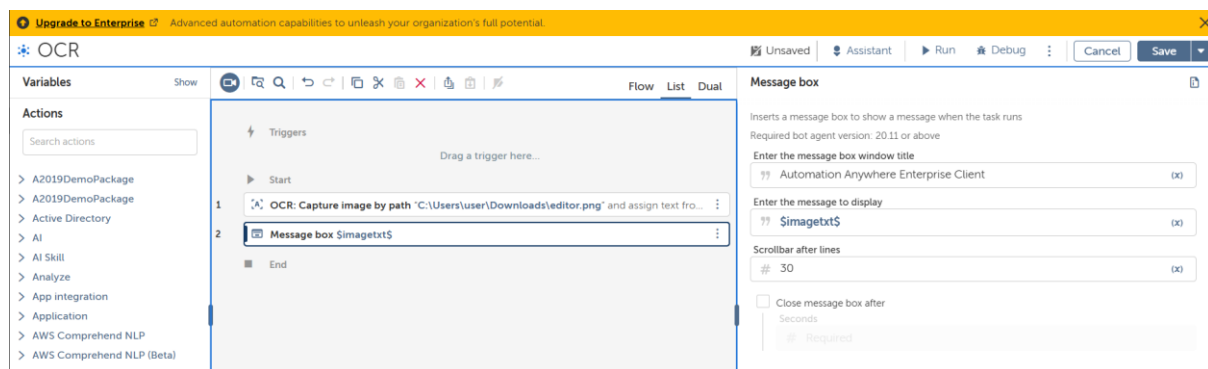


## Q7. Extracting Text form image.

1. Select OCR give path of image
2. Create variable for further use



3. Use msg to display output
4. Select variable previously created



**Input:**



**Output:**

