MSG to PDF Flow – PAD Implementation

April 2025

V1.0

Authored by: Niv Reddy

1. Introduction

Motivation and Objective

As part of my IT internship at the Fair Political Practices Commission (FPPC), I was tasked by my supervisor with finding a secure, offline method to batch convert .msg email files into .pdf format. Due to the sensitive nature of these emails, using any cloud-based or online services was strictly prohibited. This prompted a solution using Microsoft Power Automate Desktop (PAD), which offered the necessary local automation capabilities.

The objective was to create a flow that could reliably convert thousands of .msg files to PDF while being easy to use and low maintenance for other staff members. This documentation outlines the development process, logic structure, and operational flow of the automation solution.

2. Environment and Requirements

System Setup and Folder Structure

System Requirements:

- Windows 10 or 11
- Microsoft Outlook installed and configured
- Power Automate Desktop

Folder Structure:

- **InputMSG**: Source folder containing .msg files
- **OutputPDF:** Target folder for generated .pdf files
- **FailedMSG:** Folder for any .msg files that failed during conversion

Users only need to modify these three folder variables in the PAD flow before execution

3. Flow Architecture

PAD Logic Overview and Step-by-Step Flow

The flow follows this general process:

1. **File Enumeration**: Retrieve all .msg files from InputMSG.

2. **Iteration Loop**: Loop through each file and:
   - Launch the .msg file using Outlook
   - Wait for Outlook to open
   - Derive the intended PDF path using the file name
   - Check if a PDF already exists
   - If it exists: display a message and skip
   - If not: trigger the print to PDF process

3. **Print Logic:**
   - Simulate the Print keystroke (Ctrl + P)
   - UI Automation: Click the "Print" button in the dialog
   - Enter PDF path via keystroke
   - Simulate Save (Alt + F4 then Alt + N to close prompts)

4. **Failure Handling:**
   - If UI Automation fails: move .msg file to FailedMSG

5. **Cleanup:**
   - Close Outlook window for each email

4. Key Features and Enhancements

Built-in Safeguards and Improvements

- **Duplicate Handling:** Checks if the output PDF already exists and skips if found.

- **Crash Resilience:** Rare UI automation errors during printing are caught, and the failed .msg file is moved to a separate folder for review.

- **Minimal UI Automation:** Only the "Print" button click uses UI automation; all other interactions (e.g., file naming, closing) are keystroke-based to improve consistency.

- **Variable Flexibility:** Paths are abstracted into variables for easy modification.

- **Test Email Profile:** The automation uses a dedicated Outlook profile named "Fair Political Practices Commission" to ensure the exported PDFs display a clean, professional header.

5. Results and Performance

Test Run Outcome and Observations

In a controlled test run, the PAD flow successfully converted 552 .msg files to .pdf in 3 hours and 36 minutes, with only one failed conversion.

While not optimized for speed, the solution emphasizes:

- **Reliability**: Runs in the background with minimal supervision
- **Maintainability**: Logical folder structure and isolated failed files
- **Reusability**: Easily modifiable for future batches

6. Lessons Learned

Troubleshooting and Iterative Growth

- UI automation can be unreliable at scale; fallback logic is essential.
- Keystroke-based inputs offer higher consistency across repetitions.
- It’s critical to plan for mid-run terminations and duplicate entries.
- Using a dedicated Outlook profile improved output branding and minimized confusion.

.

7. Future Improvements

Feature Expansion and Automation Add-ons

- **Retry Logic:** Automate a second pass to retry files in the FailedMSG folder without manual intervention.

- **Logging:** Add timestamped logging for each processed file and reason for any failure.

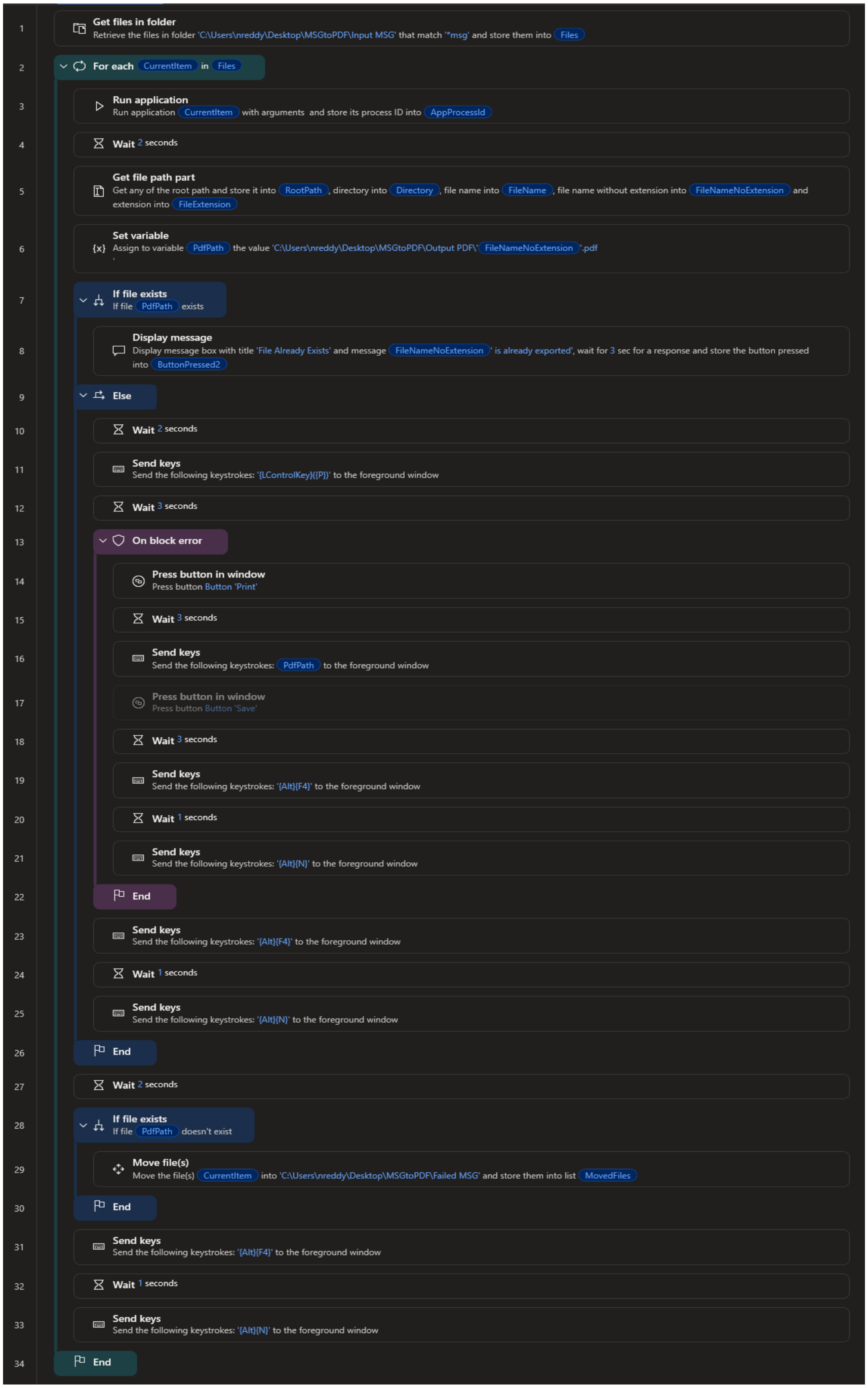
- **Metadata Extraction:** Include email metadata (e.g., subject, sender, date) in a CSV alongside the PDFs.

8. Conclusion

Project Summary and Reflection

This PAD flow provides the FPPC with a secure, offline, and reliable way to batch convert .msg files into PDFs for archival or documentation purposes. It was a hands-on learning experience in error handling, UI automation, and real-world workflow engineering.

9. Flow Screenshot



1 Get files in folder
Retrieve the files in folder 'C:\Users\nreddy\Desktop\MSGtoPDF\Input MSG' that match '*msg' and store them into Files

2 For each CurrentItem in Files

3 Run application
Run application CurrentItem with arguments and store its process ID into AppProcessId

4 Wait 2 seconds

5 Get file path part
Get any of the root path and store it into RootPath, directory into Directory, file name into FileName, file name without extension into FileNameNoExtension and extension into FileExtension

6 Set variable
Assign to variable PdfPath the value 'C:\Users\nreddy\Desktop\MSGtoPDF\Output PDF\'FileNameNoExtension'.pdf
'

7 If file exists
If file PdfPath exists

8 Display message
Display message box with title 'File Already Exists' and message FileNameNoExtension' is already exported', wait for 3 sec for a response and store the button pressed into ButtonPressed2

9 Else

10 Wait 2 seconds

11 Send keys
Send the following keystrokes: '{LControlKey}({P})' to the foreground window

12 Wait 3 seconds

13 On block error

14 Press button in window
Press button Button 'Print'

15 Wait 3 seconds

16 Send keys
Send the following keystrokes: PdfPath to the foreground window

17 Press button in window
Press button Button 'Save'

18 Wait 3 seconds

19 Send keys
Send the following keystrokes: '{Alt}{F4}' to the foreground window

20 Wait 1 seconds

21 Send keys
Send the following keystrokes: '{Alt}{N}' to the foreground window

22 End

23 Send keys
Send the following keystrokes: '{Alt}{F4}' to the foreground window

24 Wait 1 seconds

25 Send keys
Send the following keystrokes: '{Alt}{N}' to the foreground window

26 End

27 Wait 2 seconds

28 If file exists
If file PdfPath doesn't exist

29 Move file(s)
Move the file(s) CurrentItem into 'C:\Users\nreddy\Desktop\MSGtoPDF\Failed MSG' and store them into list MovedFiles

30 End

31 Send keys
Send the following keystrokes: '{Alt}{F4}' to the foreground window

32 Wait 1 seconds

33 Send keys
Send the following keystrokes: '{Alt}{N}' to the foreground window

34 End

Figure 1: Power Automate Desktop Flow Overview