Email Tracker Automation Script

# Overview

This PowerShell script automates sending follow-up emails based on data from an Excel file. It integrates with Outlook for email delivery and an Access database for tracking follow-ups. The script supports environment configuration via a `.env` file and includes logging for audit purposes.

# Features

* Loads environment variables from a `.env` file (with optional override).
* Validates required environment variables before execution.
* Reads data from an Excel sheet (`Data`).
* Groups recipients by email where Status = "Pending".
* Sends personalized HTML emails using an Outlook account.
* Attaches the Excel file if the number of computers ≥ ATTACH\_THRESHOLD.
* Adds a note in the email body when the attachment is skipped.
* Tracks follow-ups in an Access database using ACE OLEDB.
* Logs all actions and provides a summary report.

# Prerequisites

* Windows PowerShell (or PowerShell 7 with COM support).
* Microsoft Excel and Outlook installed and configured.
* Microsoft Access Database Engine (ACE OLEDB provider).
* .env file with required variables.
* Excel file with a Data sheet containing:
* - Computer Name (Column 1)
* - App Name (Column 4)
* - To (Column 5)
* - CC (Column 6)
* - Status (Column 8)

# Environment Variables

Create a `.env` file in the script directory with the following keys:

EXCEL\_PATH=C:\Path\To\Data.xlsx  
LOG\_FILE=C:\Path\To\Logs\email\_log.csv  
ACCESS\_DB\_PATH=C:\Path\To\Database.accdb  
EMAIL\_TEMPLATE\_PATH=C:\Path\To\Template.html  
DEFAULT\_SENDER=your.email@domain.com  
ATTACH\_THRESHOLD=20  
EMAIL\_DEADLINE=2025-10-15 # Optional

# How It Works

1. Load .env → Environment variables are loaded into the process.
2. Validate Variables → Script checks for required values.
3. Read Excel → Filters rows where Status = Pending.
4. Group by Recipient → Collects computers, apps, and CC addresses.
5. Compose Email → Uses HTML template and injects dynamic data.
6. Attachment Logic → Adds Excel file if computer count ≥ threshold.
7. Send Email → Uses Outlook COM object.
8. Track Follow-Up → Updates or inserts record in Access DB.
9. Log Results → Writes to CSV log file.
10. Summary → Displays counts of sent, failed, with/without attachment.

# Logging

Logs are stored in the file specified by LOG\_FILE. Format:

Timestamp,Email,Computers,Status,ErrorMessage

# Error Handling

* Missing environment variables → Script exits with an error message.
* Missing files → Script throws an error and stops.
* Database or email errors → Logged and counted in summary.

# Customization

* Modify the HTML template to adjust email content.
* Adjust ATTACH\_THRESHOLD in .env to control attachment behavior.
* Add additional placeholders in the template (e.g., {{Deadline}}).

# Execution

Run the script in PowerShell:

.\EmailTracker.ps1 -DotEnvPath "C:\Path\To\.env" -DotEnvOverride

# Summary Output

At the end of execution, the script displays:

✅ All done! Log saved to C:\Path\To\Logs\email\_log.csv  
Summary:  
 Emails Sent: X  
 Failed: Y  
 With Attachment: Z  
 Without Attachment: W

# Troubleshooting & Common Errors

## Missing Environment Variables

* Error: ❌ Missing environment variables: - EXCEL\_PATH - LOG\_FILE
* Fix: Ensure .env file exists and contains all required keys. Use -DotEnvOverride if needed.

## File Not Found

* Error: The file 'C:\Path\To\Data.xlsx' does not exist.
* Fix: Verify paths in .env and ensure files are accessible.

## Outlook COM Error

* Error: Default sender account not found in Outlook profile.
* Fix: Confirm Outlook is installed and configured with the correct account.

## ACE OLEDB Provider Not Found

* Error: Provider=Microsoft.ACE.OLEDB.12.0 not registered.
* Fix: Install Microsoft Access Database Engine.

## Excel COM Object Issues

* Symptoms: Script hangs or Excel remains open.
* Fix: Ensure proper cleanup or restart PowerShell.

## Emails Not Sending

* Possible Causes: Outlook offline or blocked.
* Fix: Open Outlook and ensure it’s connected.