

OPCUG



Users helping users
for over 40 years



Gemini &



Vibe Coding: Automating *MailStore Home* archiving

October 14, 2026

Chris Taylor

Ottawa PC Users' Group

Chris.Taylor@opcug.ca

Update MailStore archive
every day while I sleep



OPCUG



What is *MailStore Home*?

■ Email archiving program

- copy emails from a variety of mail servers
- create & maintain local archive for reading, printing, searching, restoring, moving emails to a different mail server...

https://opcug.ca/qa_recordings/20260527_PowerPoint.pdf

■ I have four email accounts

- Gmail
- OPCUG email
- National Capital Freenet email
- AOL Mail

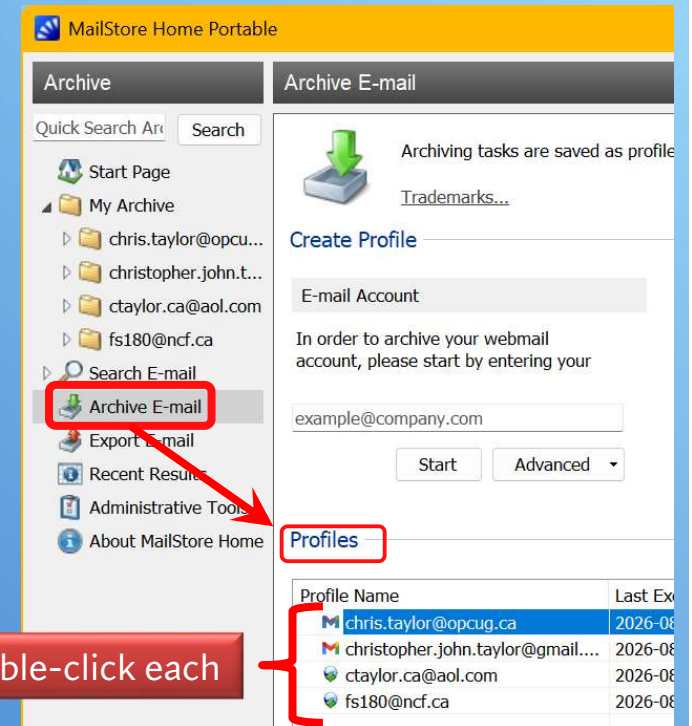
OPCUG



Users helping users
for over 40 years

Updating local archive

- Updating archive
 - double-click each archive *profile*
in my case, 4 separate archive profiles
- Want to update archive once a day
- I hate manual repetitive processes
 - tend to forget
- *No built-in way to automate updating archive*



Can vibe coding help me?

OPCUG



Users helping users
for over 40 years

Partial solution in *MailStore Home*

- Documentation for *MailStore Home* says:

- Creating a **shortcut** to run an **Archiving Profile**

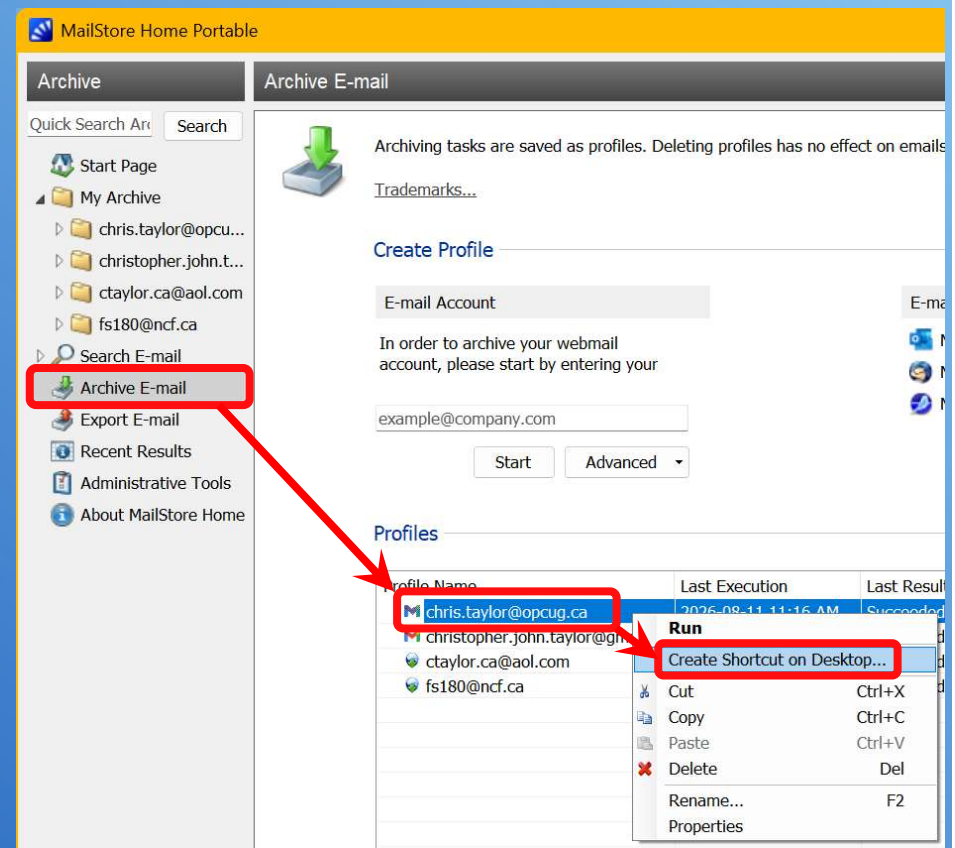
- open *MailStore Home*
- select **Archive Email** section
- right-click **profile**
- click **Create Shortcut on Desktop**

- Double-click shortcut on desktop

- *MailStore Home* updates archive for that one email account

- **Sounds like what I need**

- but need to automate running all four shortcuts

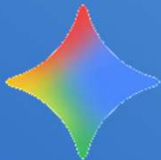


OPCUG



Users helping users
for over 40 years

What do I want vibe coding to do?

- Use *Windows PowerShell*
 - exists on every Windows computer
 - don't have to install anything
- Run all four archiving profile shortcuts
 - every day at 2:00 AM
 - I never turn my computer off
- Let's try  **Gemini** to vibe code it

Using PowerShell, run the shortcuts in c:\users\ctaylor\scripts named fs180@ncf.ca.lnk, ctaylor.ca@aol.com.lnk, christopher.john.taylor@gmail.com.lnk, and chris.taylor@opcug.ca.lnk every day at 2:00 AM.

Moved all 4 shortcuts from desktop to c:\users\ctaylor\scripts

To run these shortcuts automatically every day at 2:00 AM, the best approach is to create a small helper PowerShell script that opens them, and then schedule that script using the Windows Task Scheduler via PowerShell.

Follow these steps to set this up:

Step 1: Create the Runner Script

Create a new PowerShell script named RunShortcuts.ps1 inside your scripts folder (c:\users\ctaylor\scripts\RunShortcuts.ps1) with the following contents:

PowerShell

```
# Define the list of shortcuts
$shortcuts = @(
    "c:\users\ctaylor\scripts\fs180@ncf.ca.lnk",
    "c:\users\ctaylor\scripts\ctaylor.ca@aol.com.lnk",
    "c:\users\ctaylor\scripts\christopher.john.taylor@gmail.com.lnk",
    "c:\users\ctaylor\scripts\chris.taylor@opcug.ca.lnk"
)

# Loop through and start each shortcut
foreach ($shortcut in $shortcuts) {
    if (Test-Path $shortcut) {
        Start-Process $shortcut
        # Optional: Add a brief delay between launches if needed
        Start-Sleep -Seconds 2
    } else {
        Write-Warning "Shortcut not found: $shortcut"
    }
}
```

Gemini says what it will do

- create PowerShell script to run shortcuts
- schedule the script using Windows Task Scheduler

Click to copy script to clipboard

Save contents of clipboard as text file

- c:\users\ctaylor\scripts\RunShortcuts.ps1

Important to use names Gemini suggests

- later parts may assume that!

Step 2: Register the Scheduled Task

Open PowerShell as Administrator and run the following commands to create the automated task that triggers the script every day at 2:00 AM:

PowerShell

```
# Define task parameters
$taskName = "RunEmailShortcutsDaily"
$scriptPath = "c:\users\ctaylor\scripts\RunShortcuts.ps1"

# Create the action (runs PowerShell with the script)
$action = New-ScheduledTaskAction -Execute "PowerShell.exe" -Argument "-NoProfile -ExecutionPolicy Bypass -File \"$scriptPath\""

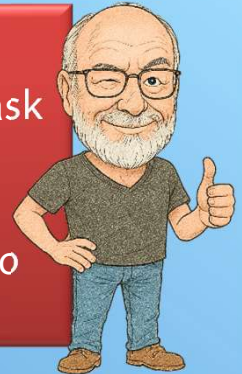
# Create the daily trigger for 2:00 AM
$trigger = New-ScheduledTaskTrigger -Daily -At "02:00AM"

# Configure settings (runs even on battery, wakes computer if needed)
$settings = New-ScheduledTaskSettingsSet -AllowStartIfOnBatteries -DontStopIfRunning -WakeToRun

# Register the task (runs under the current user account)
Register-ScheduledTask -TaskName $taskName -Action $action -Trigger $trigger -Settings $settings -User ctaylor
```

Sweet!

- other vibe-coding with a scheduled task told me steps required to *manually* create a scheduled task
- Gemini gave me a PowerShell script to create the task



Administrator: Windows PowerShell

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

```
PS C:\WINDOWS\system32> # Define task parameters
> $taskName = "RunEmailShortcutsDaily"
> $scriptPath = "c:\users\ctaylor\scripts\RunShortcuts.ps1"
>
> # Create the action (runs PowerShell with the script)
> $action = New-ScheduledTaskAction -Execute "PowerShell.exe" -Argument "-NoProfile -ExecutionPolicy Bypass -File \"$scriptPath\""
>
> # Create the daily trigger for 2:00 AM
> $trigger = New-ScheduledTaskTrigger -Daily -At "02:00AM"
>
> # Configure settings (runs even on battery, wakes computer if needed)
> $settings = New-ScheduledTaskSettingsSet -AllowStartIfOnBatteries -DontStopIfRunning -WakeToRun
>
> # Register the task (runs under the current user account)
> Register-ScheduledTask -TaskName $taskName -Action $action -Trigger $trigger -Settings $settings -User ctaylor
```

Register-ScheduledTask : A parameter cannot be found that matches parameter name 'Action\$action'.
At line:15 char:44

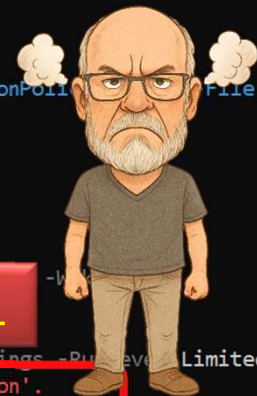
+ Register-ScheduledTask -TaskName \$taskName -Action\$action -Trigger \$t ...

+ ~~~~~

+ CategoryInfo : InvalidArgument: (:) [Register-ScheduledTask], ParameterBindingException
+ FullyQualifiedErrorId : NamedParameterNotFound,Register-ScheduledTask

PS C:\WINDOWS\system32>

Error #1



when doing step 2, error: Register-ScheduledTask : A parameter cannot be found that matches parameter name 'Action\$action'.

At line:15 char:44

```
+ Register-ScheduledTask -TaskName $taskName -  
Action$action -Trigger $t ...  
+  
+ CategoryInfo          : InvalidArgument: (:) [Register-  
ScheduledTask], ParameterBindingException  
+ FullyQualifiedErrorId :  
NamedParameterNotFound,Register-ScheduledTask
```

Pasted in error message

*Hey, Gemini
you wrote the code!*

Ah, that error happened because there is a missing space between `-Action` and `$action`.

Here is the corrected block of code for Step 2. Just copy and paste this into your PowerShell window:

PowerShell

```
# Define task parameters  
$taskName = "RunEmailShortcutsDaily"  
$scriptPath = "c:\users\ctaylor\scripts\RunShortcuts.ps1"  
  
# Create the action (runs PowerShell with the script)  
$action = New-ScheduledTaskAction -Execute "PowerShell.exe" -Argument "-NoProf:  
  
# Create the daily trigger for 2:00 AM  
$trigger = New-ScheduledTaskTrigger -Daily -At "02:00AM"  
  
# Configure settings (runs even on battery, wakes computer if needed)  
$settings = New-ScheduledTaskSettingsSet -AllowStartIfOnBatteries -DontStopIfGo  
  
# Register the task (runs under the current user account)  
Register-ScheduledTask -TaskName $taskName -Action$action -Trigger $trigger -Setting
```

Tried suggested replacement script and got same error

Error #2

Noticed the replacement script has the same

missing space between `-Action` and `$action`

mistake!



Could manually edit the script

- but let's not let Gemini off the hook

I still get error: Register-ScheduledTask : A parameter cannot be found that matches parameter name 'Action\$action'.

At line:15 char:44

+ Register-ScheduledTask -TaskName \$taskName -

Pasted in error message

What ???

This error happens when text gets slightly joined together during a copy-and-paste action.

Run this single-line command instead, which avoids spacing issues by nesting the commands directly:

Go with the flow

PowerShell

```
Register-ScheduledTask -TaskName "RunEmailShortcutsDaily" -Action (New-ScheduledTaskAction -Execute "PowerShell.exe" -Argument "-NoProfile -ExecutionPolicy Bypass -File `\"c:\users\ctaylor\scripts\RunShortcuts.ps1`\"") -Trigger (New-ScheduledTaskTrigger -Daily -At "02:00AM") -Settings (New-ScheduledTaskSettingsSet -AllowStartIfOnBatteries -DontStopIfGoingOnBatteries -WakeToRun) -RunLevel Limited
```

No error message

Administrator: Windows PowerShell

```
PS C:\WINDOWS\system32> Register-ScheduledTask -TaskName "RunEmailShortcutsDaily" -Action (New-ScheduledTaskAction -Execute "PowerShell.exe" -Argument "-NoProfile -ExecutionPolicy Bypass -File `\"c:\users\ctaylor\scripts\RunShortcuts.ps1`\"") -Trigger (New-ScheduledTaskTrigger -Daily -At "02:00AM") -Settings (New-ScheduledTaskSettingsSet -AllowStartIfOnBatteries -DontStopIfGoingOnBatteries -WakeToRun) -RunLevel Limited
```

TaskPath	TaskName	State
\\	RunEmailShortcutsDaily	Ready

```
PS C:\WINDOWS\system32>
```



Verification

To verify that the task was successfully registered, run:

PowerShell

```
Get-ScheduledTask -TaskName "RunEmailShortcutsDaily"
```

If successful, it will return the task details with a State of Ready.

Success: scheduled task has been created

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> Get-ScheduledTask -TaskName "RunEmailShortcutsDaily"

TaskPath          TaskName          State
-----
\                  RunEmailShortcutsDaily Ready

PS C:\WINDOWS\system32> _
```



Managing or Testing Your Task

- **Test Immediately:** You can test the setup right now without waiting for 2:00 AM by running:

PowerShell

```
Start-ScheduledTask -TaskName "RunEmailShortcutsDaily"
```

Shouldn't Gemini have known this?



Just fix the problem

Updated the script and used above to *Test Immediately*

Error #3

- a window flashed & disappeared
- email not archived in MailStore

when I run the PowerShell command from the "Test immediately" section above, a new window flashes and goes away immediately, rather than running the programs in the shortcuts

When you use `Start-ScheduledTask`, it runs the task in the background using the SYSTEM account or a hidden non-interactive session. Because of this, any graphical apps or browser windows launched by shortcuts (like email links) won't show up on your desktop screen—they are opening in an invisible session.

To fix this and make the shortcuts open properly on your actual desktop, update your script to launch them using the interactive desktop session.

Update the RunShortcuts.ps1 Script

1 min

Open your `c:\users\ctaylor\scripts\RunShortcuts.ps1` file and replace its contents with the following code, which targets your interactive desktop session:

PowerShell

```
$shortcuts = @(
    "c:\users\ctaylor\scripts\fs180@ncf.ca.lnk",
    "c:\users\ctaylor\scripts\ctaylor.ca@aol.com.lnk",
    "c:\users\ctaylor\scripts\christopher.john.taylor@gmail.com.lnk",
    "c:\users\ctaylor\scripts\chris.taylor@opcug.ca.lnk"
)

foreach ($shortcut in $shortcuts) {
    if (Test-Path $shortcut) {
        # Use explorer.exe to force open the shortcut in the active desktop user
        Start-Process "explorer.exe" -ArgumentList "" $shortcut ""
        Start-Sleep -Seconds 3
    } else {
        Write-Warning "Shortcut not found: $shortcut"
    }
}
```

Verification: Save the file and run `"c:\users\ctaylor\scripts\RunShortcuts.ps1"` directly in PowerShell to confirm the windows now open up on your screen.

It worked!

- *MailStore Home* ran



MailStore Home Portable

Start Page

Type Your Name Here
10,588 messages archived
Version 26.3

MailStore
by opentext

Quick Access

- Archive E-mail
- Search E-mail
- Export E-mail
- Backup to HDD or USB
- Password Protection
- Administrative Tools

Dashboard

MailStore Home

Version 26.3.0.24111 (Build Date 2016-03-01)

Tell Others About MailStore

Archive

Total Size 10,588 messages / 786 MB

Location Z:\MailStore Home

Available Disk Space 2.57 TB

License

MailStore Home is free for non-commercial usage.

Helpful Links

- MailStore Home Help
- MailStore Home Website
- MailStore Home Support Community
- MailStore on Facebook
- MailStore Business Products

Featured Products & Content

Are you an IT professional? Try MailStore Server for SMBs. **Test now**

Progress View

✓	chris.taylor@opcug.ca	The process completed successfully. 2,978 items processed.	Details...
✓	christopher.john.taylor@gmail.com	The process completed successfully. 3,846 items processed.	Details...
✓	ctaylor.ca@aol.com	The process completed successfully. 14 items processed.	Details...
✓	fs180@ncf.ca	The process completed successfully. 175 items processed.	Details...

[Clear List](#) [Close](#)

- and updated archive for all four email accounts

Might be nice to get a permanent record

Modify the script so that every time it runs it adds to a file called log.txt in the scripts folder a line:
MailStore archived:
followed by the date and time. Always add the latest line to the top of the file.

To update your script to log every run with the date and time added to the **top** of log.txt, run the following command in PowerShell. This will overwrite RunShortcuts.ps1 with the updated logic:

Error #4

- updated script failed completely!
- futzed around a bunch telling Gemini about error details



PowerShell

```
$logPath = "c:\users\ctaylor\scripts\log.txt"
$timestamp = Get-Date -Format "yyyy-MM-dd HH:mm:ss"
$newLogEntry = "MailStore archived: $timestamp"

if (Test-Path $logPath) {
    # Read existing log contents, prepend the new entry, and save back
    $existingLogs = Get-Content $logPath
    $updatedLogs = @($newLogEntry) + $existingLogs
    Set-Content -Path $logPath -Value $updatedLogs -Encoding UTF8
} else {
    # Create the log file if it doesn't exist yet
    Set-Content -Path $logPath -Value $newLogEntry -Encoding UTF8
}
```

Final portion of code doing logging

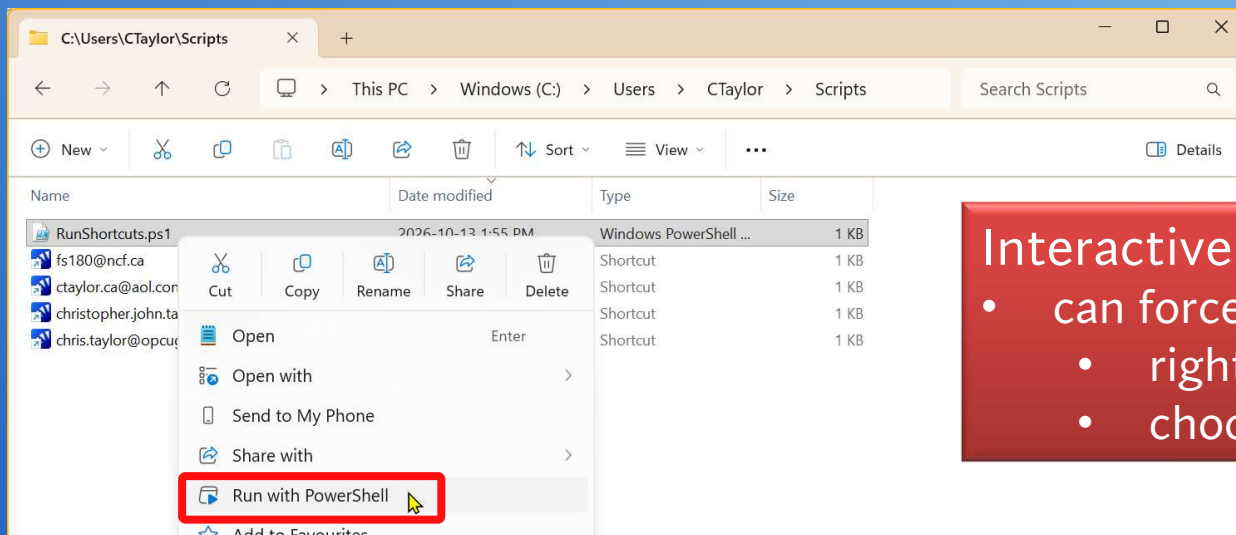
Log file

```
log.txt
File Edit View H1 B I S ...
MailStore archived: 2026-10-14 02:00:14
MailStore archived: 2026-10-13 15:39:43
MailStore archived: 2026-10-13 15:08:43
MailStore archived: 2026-10-13 15:08:04
Ln 8, Col 1 163 characters Plain text 100% Windows (CRLF) UTF-8 with BOM
```

End result

Scheduling works

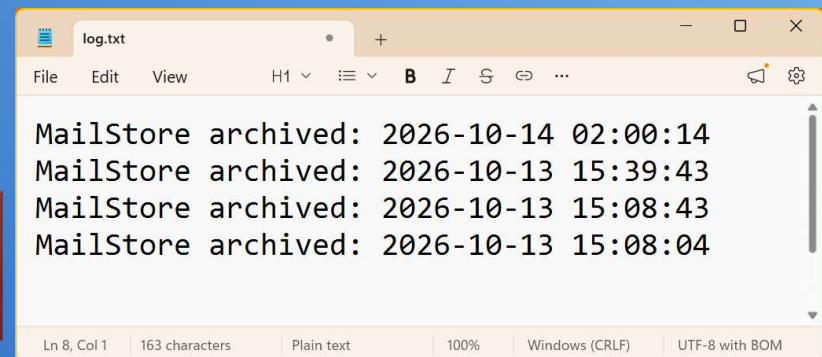
- MailStore Home updates archive of all four email accounts every day at 2:00 AM



Interactive updating

- can force an update at any time
 - right-click script
 - choose *Run with PowerShell*

Log file keeps record of when updating occurred





What I learned about vibe coding

- You don't **need** to know **anything** about programming
 - non-programmers **can** create programs
- AI **will** make mistakes
 - mistakes are not necessarily easy to identify
 - directing AI to fix mistakes works
 - but can be **very tedious** and **time consuming**
- The more you have to iterate
 - the more difficult it may be to modify the project later
 - might be easier to start over!
- But I think vibe coding makes it
 - **possible** for non-programmers to create programs
 - **easier** for experienced programmers to create programs

