



Placement Empowerment Program

Cloud Computing and DevOps Centre

Automate File Copying with a Script

Create a script to copy files from one folder to another automatically.

Name: Vishnu K

Department: CSE

Introduction

Automating repetitive tasks like file copying can save time, reduce errors, and improve efficiency in various scenarios, such as managing backups, organizing files, or syncing data between directories. This POC script will demonstrate how to copy files automatically from one folder to another, using a simple yet effective approach.

Overview

The script automates the process of copying files from a source folder to a destination folder. It ensures that all files in the source directory are transferred efficiently while maintaining their integrity. This automation can be customized based on specific needs, such as copying files based on file type, modification date, or size. The script can be written using programming languages like Python, Shell scripting, or PowerShell, depending on the operating environment.

Objectives

The key objectives of this automation task are:

Simplify File Management: Reduce manual effort in moving or copying files between folders.

Improve Efficiency: Automate the process to save time and improve productivity.

Ensure Data Integrity: Accurately copy files without any corruption or loss.

Customizable Workflow: Enable flexibility to meet specific requirements (e.g., filtering files by type or date).

Scalable Solution: Allow the script to handle a large number of files efficiently.

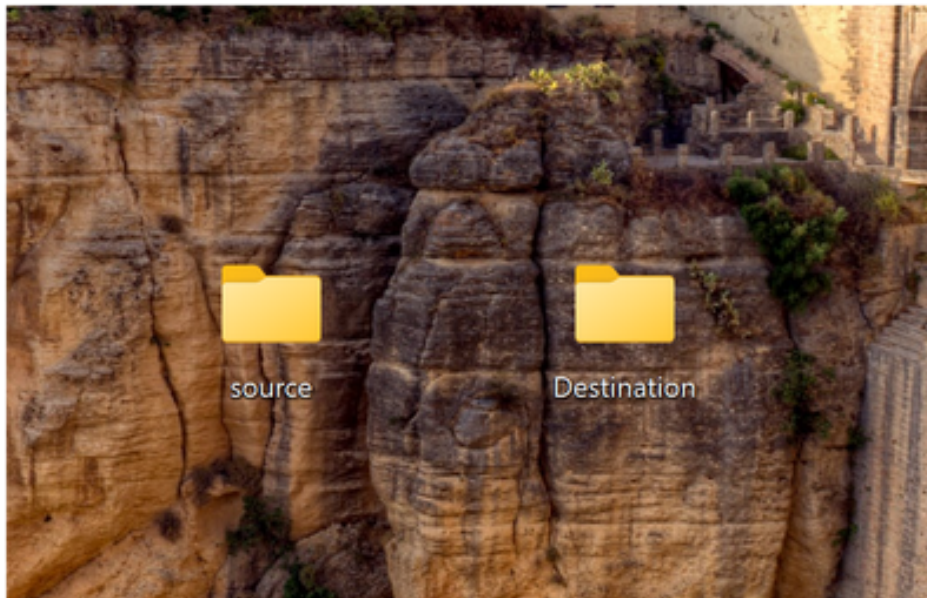
Importance

- 1. Time-Saving:** Automating the process eliminates the need for manual intervention, saving hours of work.
- 2. Error Reduction:** Reduces human errors, such as forgetting to copy specific files or overwriting important ones.
- 3. Improved Organization:** Helps maintain a consistent structure for file storage and backups.
- 4. Useful in Various Scenarios:** This approach is valuable for IT professionals, businesses, and individuals dealing with frequent file transfers or backups.
- 5. Scalability and Reusability:** Once created, the script can be reused and scaled up to handle more complex tasks.

Step-by-Step Overview

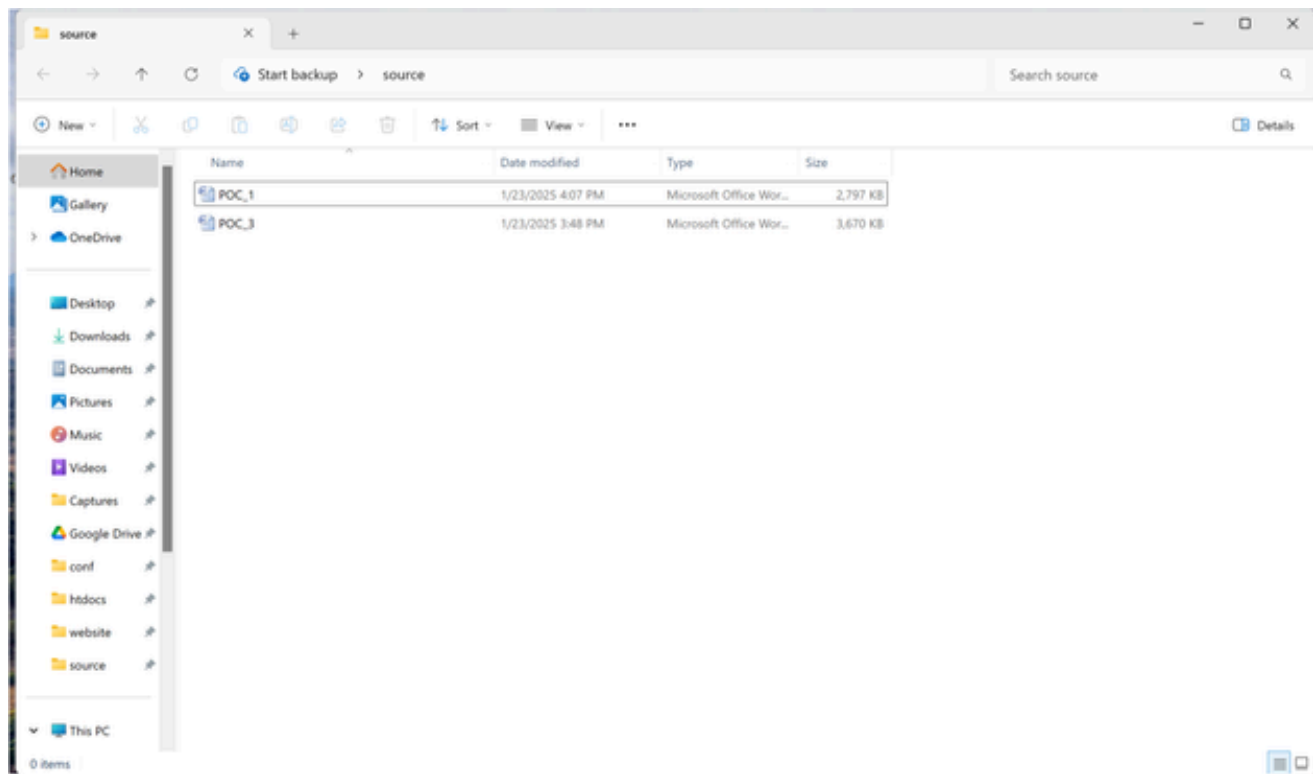
Step 1

Create two folders named Source and Destination



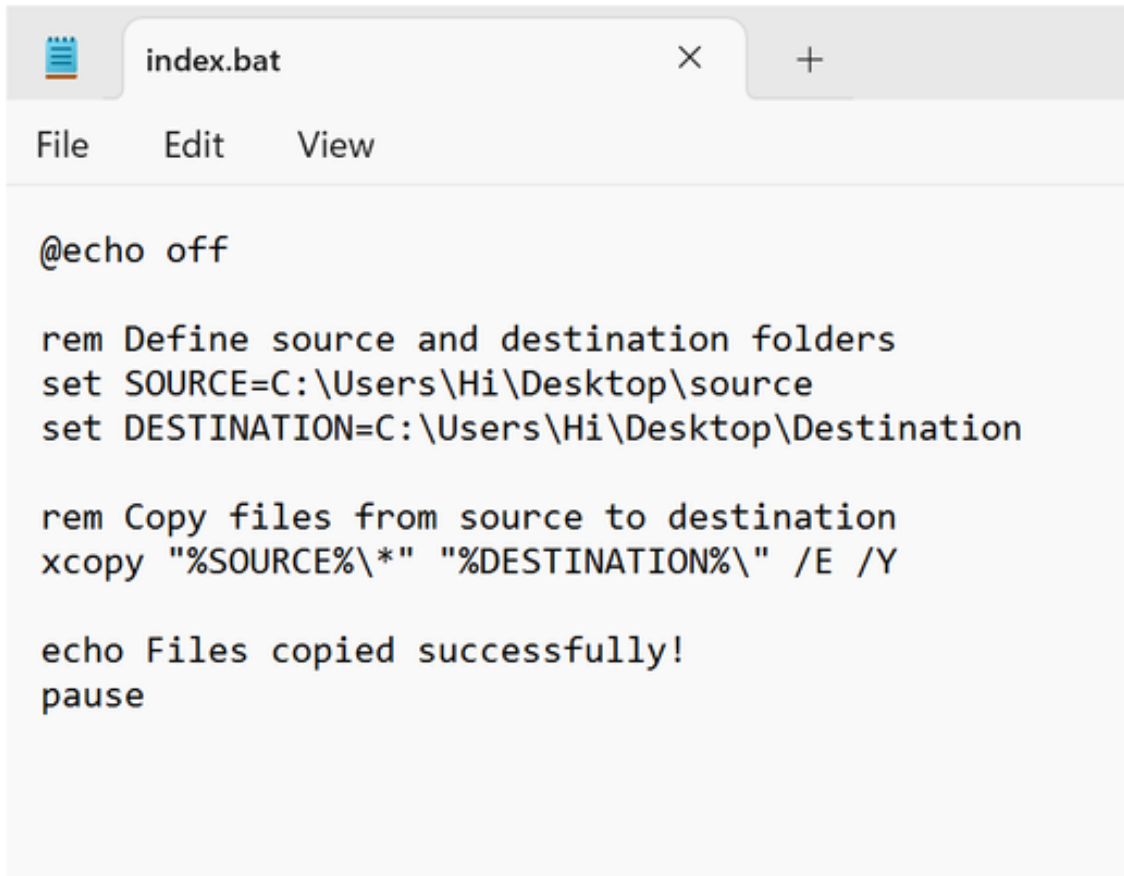
Step 2

Store some files inside it to automate it



Step 3

Open the note pad and type the code and make sure that in set **SOURCE** give your Source folder address and in set **DESTINATION** give your Destination Folder address



```
@echo off

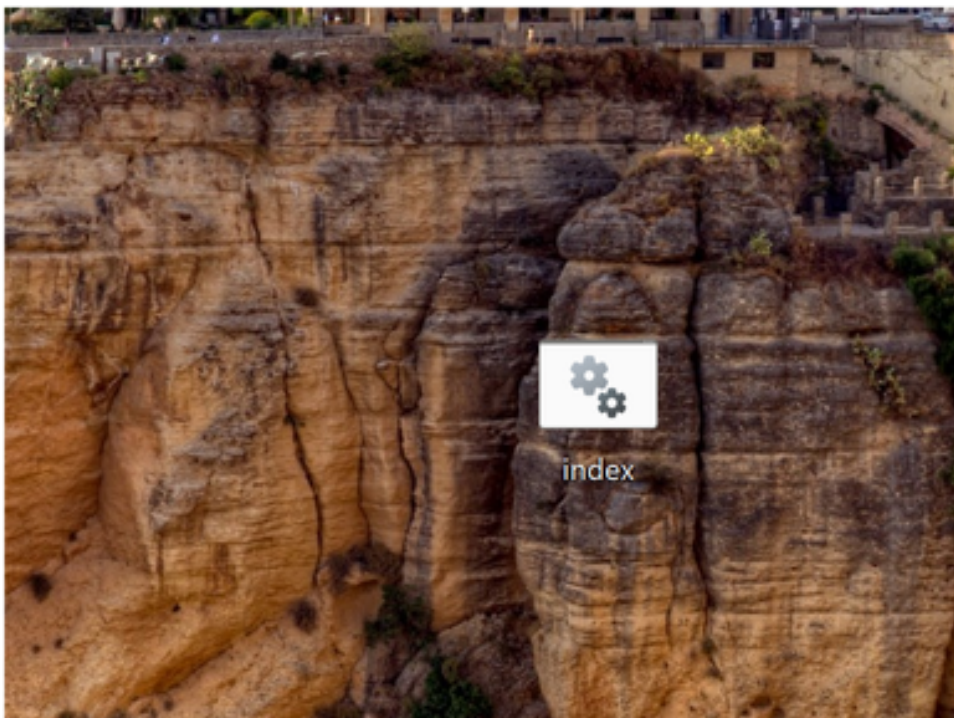
rem Define source and destination folders
set SOURCE=C:\Users\Hi\Desktop\source
set DESTINATION=C:\Users\Hi\Desktop\Destination

rem Copy files from source to destination
xcopy "%SOURCE%\*" "%DESTINATION%" /E /Y

echo Files copied successfully!
pause
```

Step 4

Then save the file in desktop with .bat extension (eg : index.bat) so the file looks like this



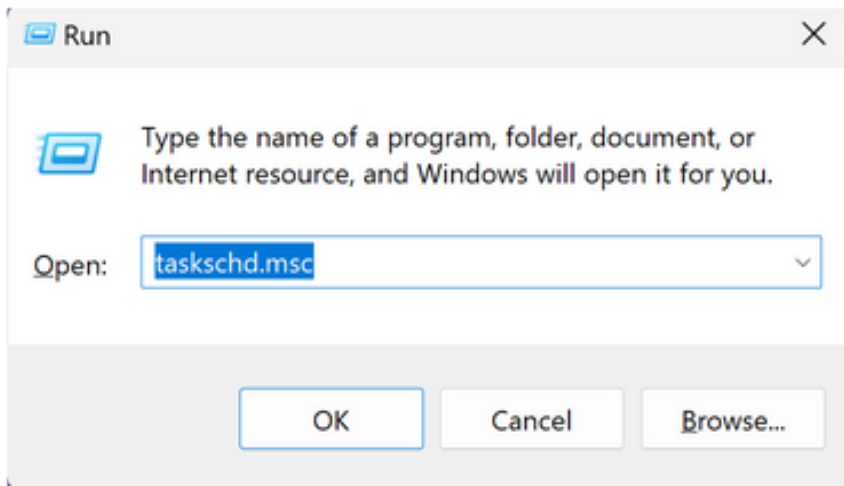
Step 5

Press **Win + R** on your keyboard.

A small "Run" dialog box will pop up.

Type **taskschd.msc** (without quotes) in the Run box.

Press Enter or click OK. This will open the Task Scheduler window.

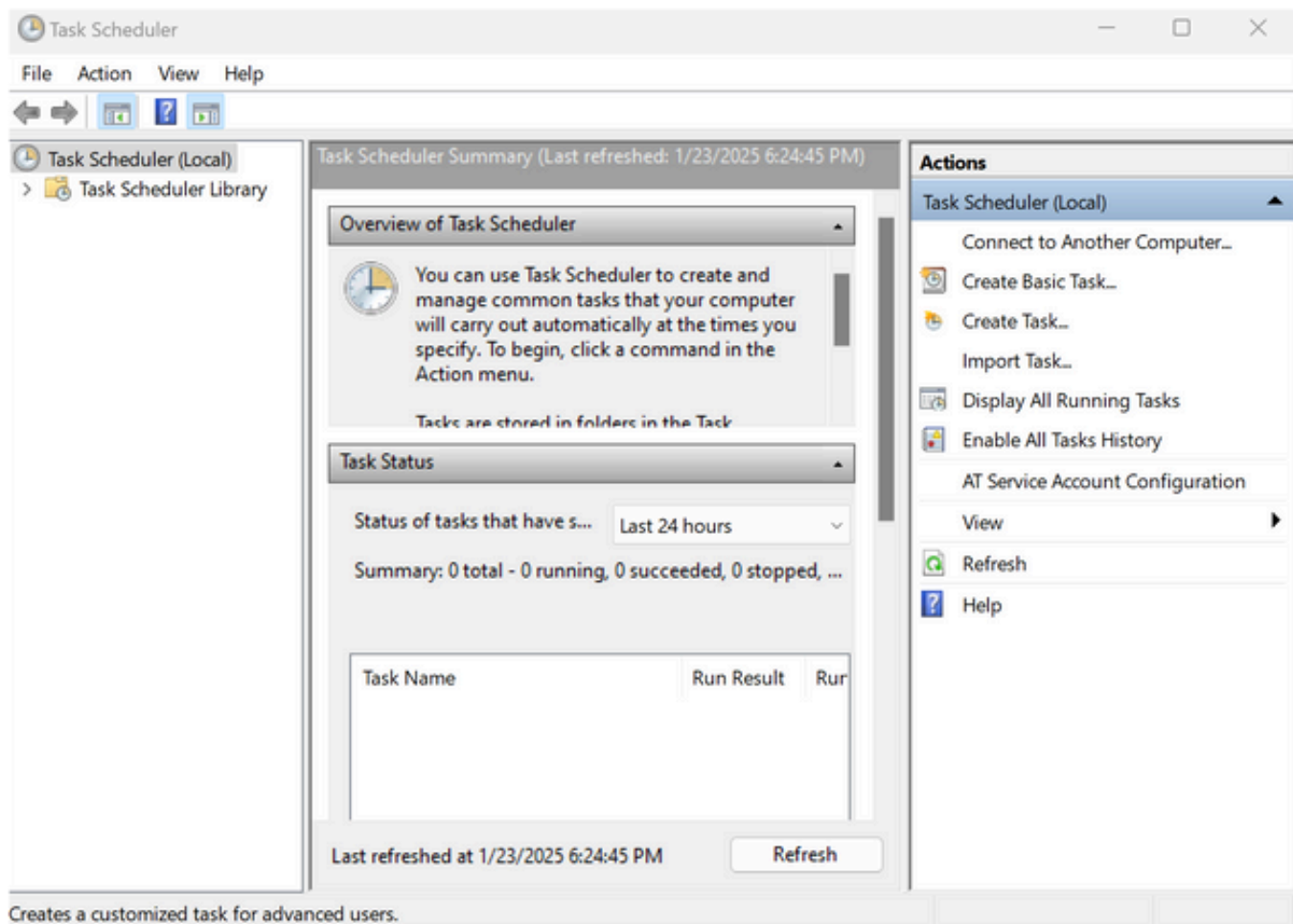


Step 6

In the Task Scheduler window, look to the right-hand side for a button called "Create Basic Task".

Click it.

A wizard will open to guide you through the setup.



Step 7

1. Enter a Name for the Task:

For example: "Automate File Copying".(This can be anything that helps you remember what the task does.)

Optionally, you can add a description like "Copies files from folder A to folder B".

1. Click Next to continue.

Create Basic Task Wizard

Create a Basic Task

Use this wizard to quickly schedule a common task. For more advanced options or settings such as multiple task actions or triggers, use the Create Task command in the Actions pane.

Name: Automate File Copying

Description: Copies files from folder A to folder B

< Back Next > Cancel

Step 8

Choose a Schedule:

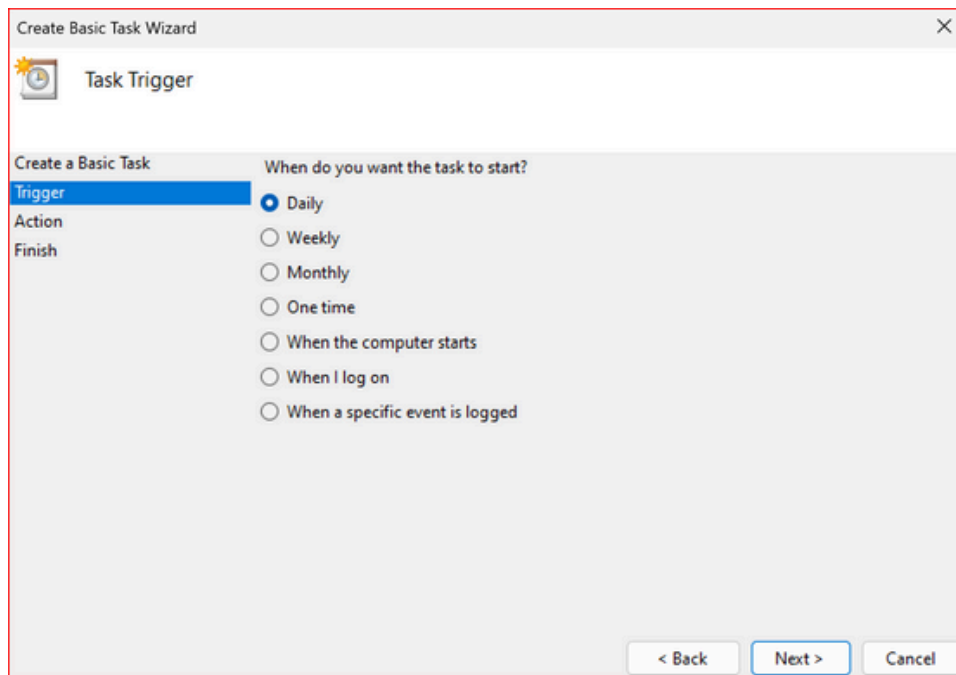
You will see options like:

Daily (runs every day).

Weekly (runs once a week).

One time (runs only once at a specific time).

Choose what works for you (e.g., Daily) and click Next.



Step 9

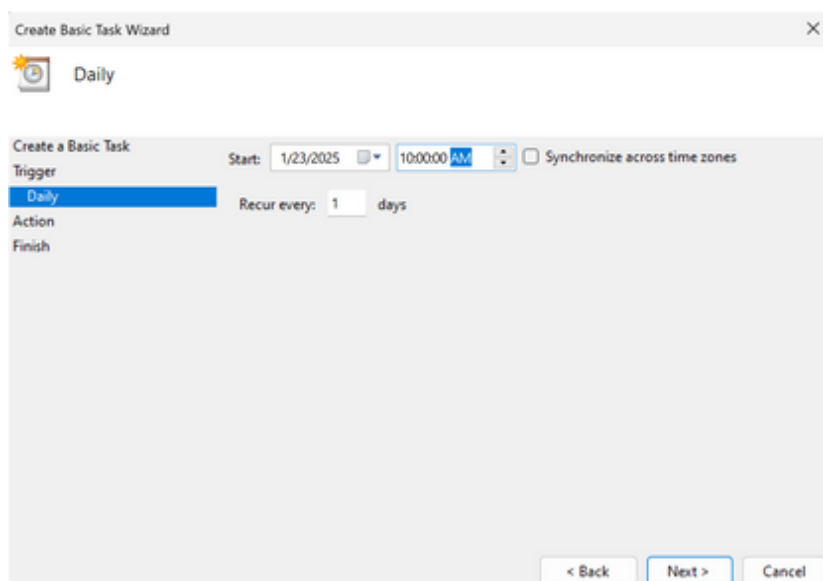
Set the Time and Frequency:

If you chose Daily, specify:

The start date (it defaults to today).

The time (e.g., 10:00 AM).

Click Next to move on.



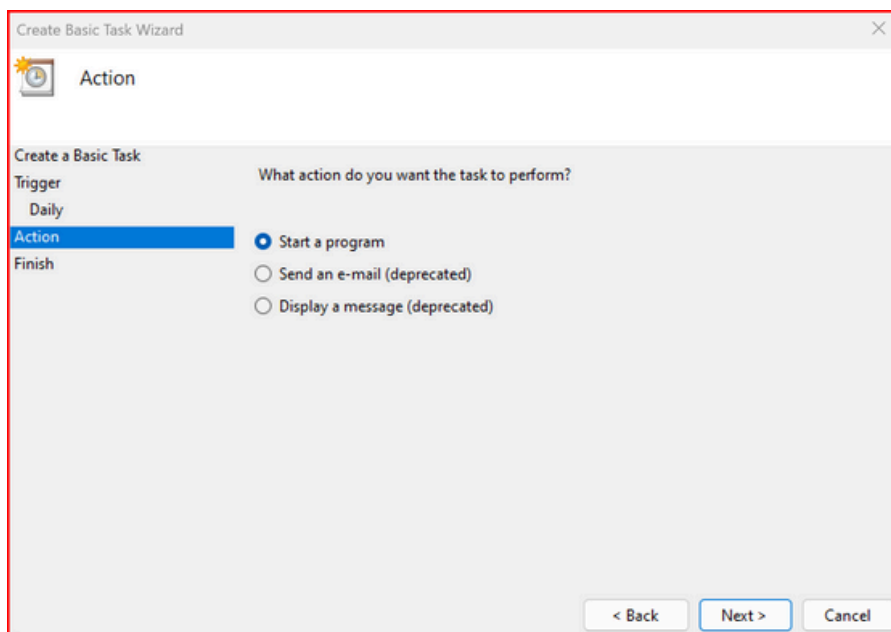
Step 10

Set the Action

Now, we tell Task Scheduler what to do when it runs.

Select "Start a Program":

On the "Action" screen, select the option "Start a Program" and click Next.



Step 11


Point to the Program or Script:

In the Program/script field, click **Browse** and navigate to the location of your .bat file.

Example: If your script is named index.bat and saved on the desktop, navigate to that file and select it.

Click Next.

Create Basic Task Wizard

 Start a Program

Create a Basic Task

Trigger

Daily

Action

Start a Program

Finish

Program/script:

C:\Users\Hi\Desktop\index.bat

Browse...

Add arguments (optional):

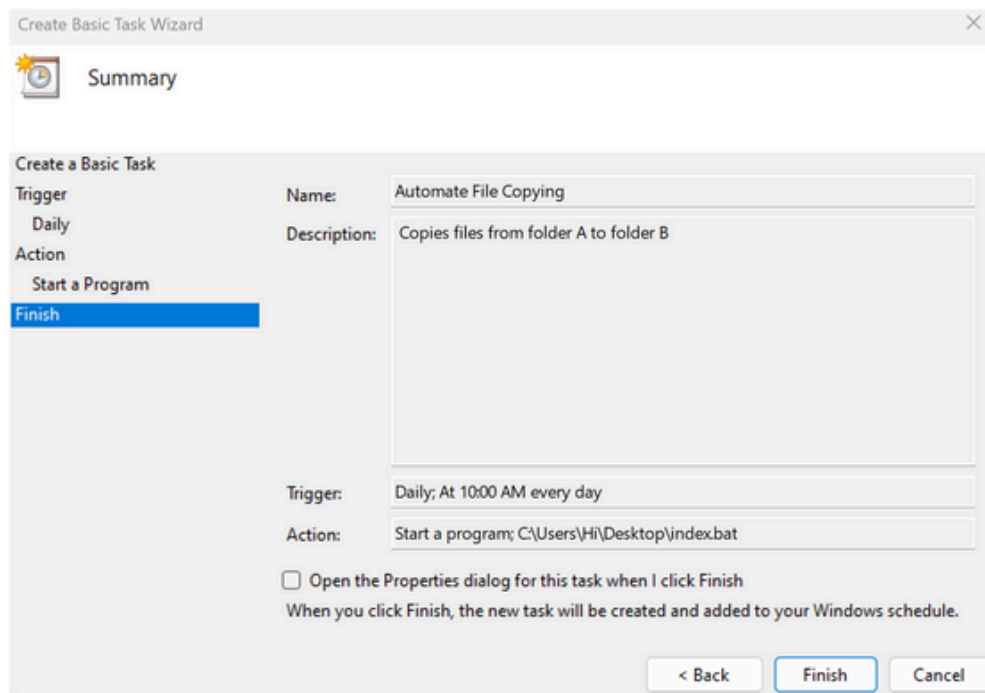
Start in (optional):

< Back Next > Cancel

Step 12

Review and Finish

Click **Finish** to save and schedule the task.

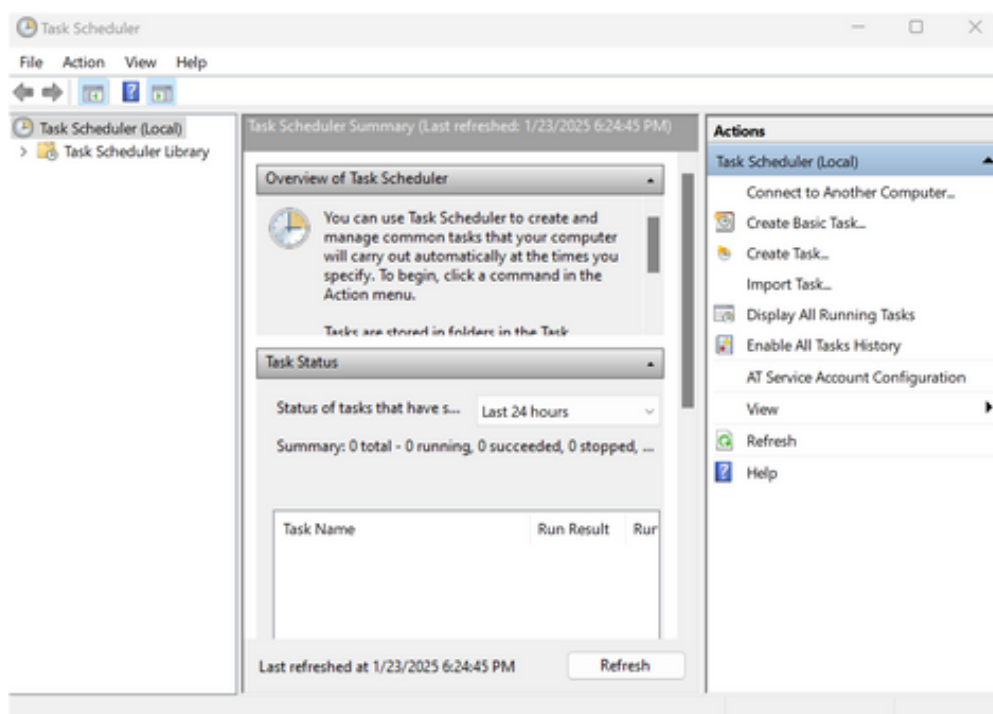


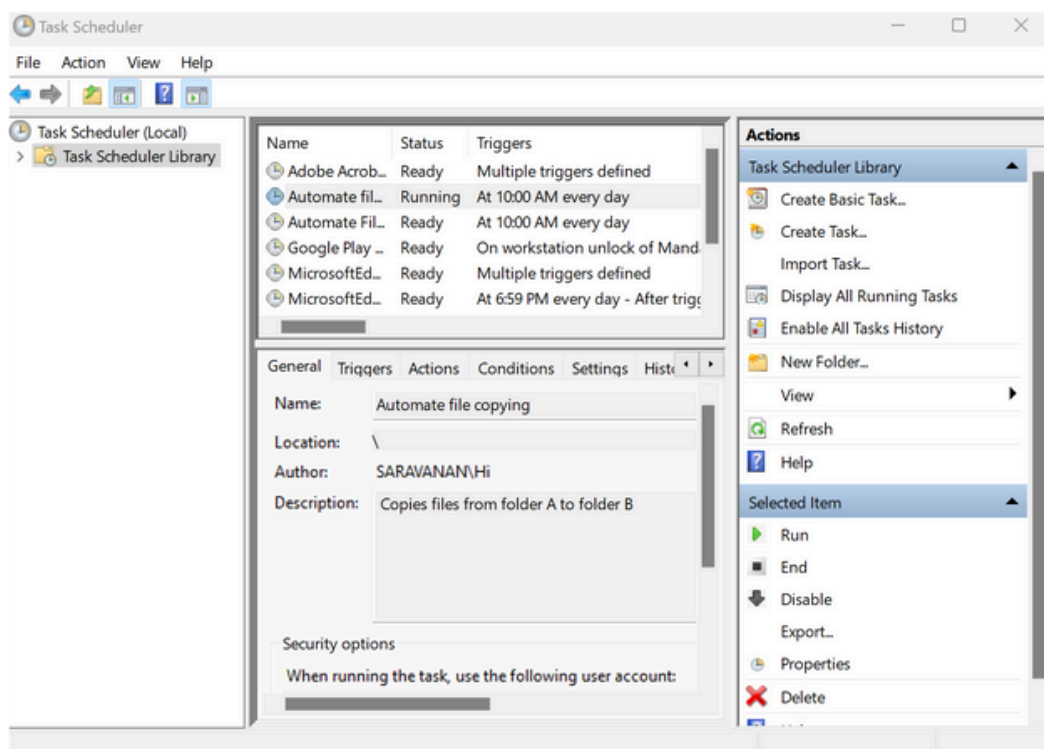
Step 13

In Task Scheduler, go to the **Task Scheduler Library** (on the left-hand side). Find your task (it should have the name you gave it, e.g., "Automate File Copying").

Right-click the task and select **Run**.

This will manually trigger the task immediately.

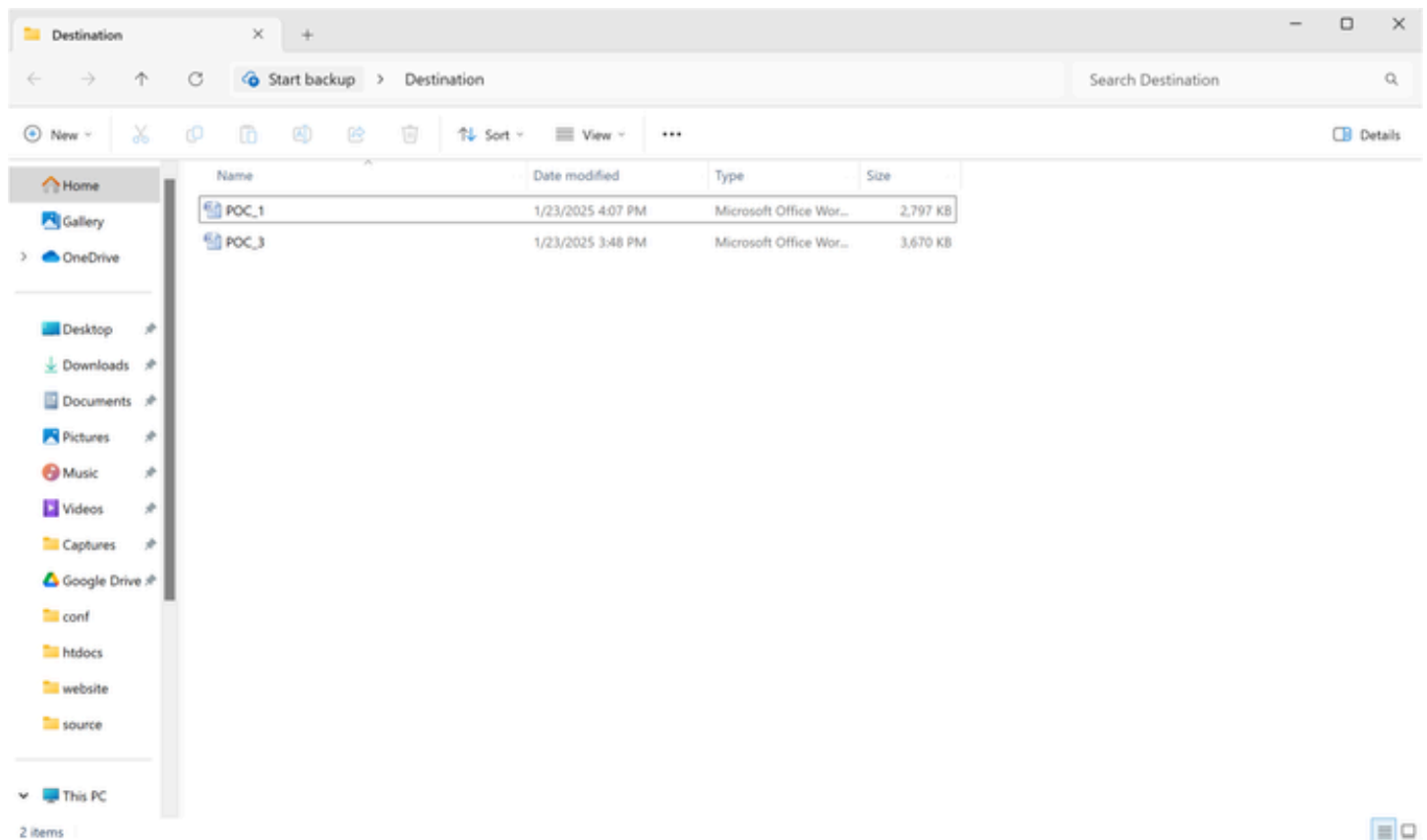




```
C:\Windows\SYSTEM32\cmd.exe
C:\Users\Hi\Desktop\source\POC_1.docx
C:\Users\Hi\Desktop\source\POC_3.docx
2 File(s) copied
Files copied successfully!
Press any key to continue . . . |
```

Step 14

If your task was set up to copy files, go to the destination folder and confirm that the files have been copied.



Outcome

By completing this Proof of Concept (PoC) of automating a task using Task Scheduler, you will:

1. Successfully set up an automated task that triggers on a schedule or when manually run.
2. Execute a batch script to copy files from a source folder to a destination folder automatically.
3. Understand how to use Task Scheduler to automate repetitive tasks in Windows.
4. Gain familiarity with task triggers, actions, and conditions in Task Scheduler.
5. Save time and ensure consistent execution of file operations without manual intervention.
6. Optionally test the task to verify that it runs as expected and achieves the desired outcome.

7. Enhance your workflow automation skills with practical hands-on experience.