 

**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: MIRDULA C Department: ADS



**Introduction**

### Automating repetitive tasks, such as file copying, enhances efficiency, minimizes errors, and saves time in various scenarios, including backup management, file organization, and data synchronization. This proof of concept (POC) script will demonstrate a streamlined approach to automatically copying files from one folder to another.

### ****Overview****

This script automates the process of copying files from a source folder to a destination folder, ensuring efficient transfer while preserving file integrity. It can be customized to meet specific requirements, such as filtering files by type, modification date, or size. Depending on the operating environment, the script can be implemented using programming languages like Python, Shell scripting, or PowerShell.

**Objectives**

**Key Objectives of This Automation Task**

* **Simplify File Management:** Minimize manual effort in transferring files between directories.
* **Enhance Efficiency:** Automate the process to save time and boost productivity.
* **Ensure Data Integrity:** Maintain file accuracy and prevent corruption or loss during transfer.
* **Customizable Workflow:** Adapt the automation to specific needs, such as filtering files by type, date, or size.
* **Scalable Solution:** Enable the script to efficiently handle large volumes of files.

**Importance**

Benefits of Automating File Copying

Time Efficiency: Eliminates manual intervention, significantly reducing workload and saving valuable time.

Error Reduction: Minimizes human errors, such as missing files or accidental overwrites.

Enhanced Organization: Maintains a structured and consistent approach to file storage and backups.

Versatility: Applicable across various scenarios, benefiting IT professionals, businesses, and individuals managing frequent file transfers or backups.

Scalability and Reusability: Designed for long-term use, allowing easy adaptation and expansion for more complex automation tasks.

**Step-by-Step Overview**

Steps to Automate File Copying with Task Scheduler

**Step 1**: Prepare the Folders

Create two folders named Source and Destination on your system.

**Step 2**: Add Files for Automation

Store some files in the Source folder that you wish to automate the copying of.

**Step 3**: Write the Script

Open Notepad and write the automation script.

In the script, set the SOURCE variable to the path of your Source folder and the DESTINATION variable to the path of your Destination folder.

**Step 4**: Save the Script

Save the script to your desktop with a .bat extension (e.g., index.bat).

**Step 5**: Open Task Scheduler

Press Win + R to open the Run dialog box.

Type taskschd.msc and press Enter or click OK to open the Task Scheduler window.

**Step 6**: Create a New Task

In the Task Scheduler window, click on Create Basic Task on the right-hand side.

This will open a wizard to guide you through the task setup.

**Step 7**: Name the Task

Enter a name for the task, such as "Automate File Copying" (you can choose any name that helps you identify the task).

Optionally, provide a description, such as "Copies files from folder A to folder B".

Click Next to continue.

**Step 8**: Choose a Schedule

Select how often you want the task to run:

Daily (runs every day)

Weekly (runs once a week)

One time (runs once at a specific time)

Choose the schedule that works best for you (e.g., Daily) and click Next.

**Step 9**: Set the Time and Frequency

If you selected Daily, specify the start date and time (e.g., 10:00 AM).

Click Next to proceed.

**Step 10**: Define the Action

Select "Start a Program" as the action to be performed by the task.

Click Next.

**Step 11**: Locate the Script

In the Program/script field, click Browse and navigate to the location of your .bat file (e.g., index.bat located on the desktop).

Select the script and click Next.

**Step 12**: Review and Finish

Review the task details and click Finish to save and schedule the task.

**Step 13**: Manually Trigger the Task

In Task Scheduler, go to the Task Scheduler Library (on the left-hand side).

Find the task you created (e.g., "Automate File Copying").

Right-click the task and select Run to manually trigger the task immediately.

**Step 14**: Verify the File Copying Process

Navigate to the Destination folder and confirm that the files have been successfully copied from the Source folder.

**Outcome**

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

1. Set up an automated task that triggers either on a schedule or when manually initiated.
2. Execute a batch script to automatically copy files from a source folder to a destination folder.
3. Learn how to use **Task Scheduler** to automate repetitive tasks in Windows.
4. Gain hands-on experience with task triggers, actions, and conditions in **Task Scheduler**.
5. Save time and ensure consistent execution of file operations without manual intervention.
6. Optionally verify the task’s functionality by testing it and confirming the desired outcome.
7. Enhance your workflow automation skills through practical, real-world experience.