**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: Kirubalini S I Department: IT



**Introduction**

### In many workflows, the need to copy files from one folder to another arises frequently. Manually copying files can be time-consuming and prone to error, especially when dealing with large numbers of files. Automating this process with a script can save time and ensure consistency. By writing a simple script, you can set up automatic file copying, streamlining your work and reducing the chances of missing or misplaced files. This guide will walk you through the steps of creating a script that automates the process of copying files from one folder to another efficiently and reliably.

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

Automating the process of copying files between folders can significantly improve efficiency, especially when handling large amounts of data. Writing a script to copy files automatically eliminates the need for repetitive manual tasks and ensures that files are consistently transferred to the correct location. Whether you're managing backups, organizing documents, or simply need to move files regularly, a custom script can handle the job for you.

In this guide, we'll explore how to create a simple script that automates the process of copying files from one folder to another. The script can be easily customized to suit your specific needs, helping you save time and reduce the risk of errors.

**Objectives**

The key objectives in **"Automate File Copying with a Script"** and **"Create a script to copy files from one folder to another automatically"** include:

**Automation of repetitive tasks:** To eliminate the need for manually copying files every time, saving time and effort.

**Efficiency:** By using a script, file copying can be done faster and without errors that often occur with manual processes.

**Consistency:** Ensuring that the correct files are copied every time and reducing the chance of missing files or performing incorrect actions.

**Customization:** Allowing users to modify the script based on their specific needs (e.g., specifying source and destination directories, handling different file types, etc.).

**Error reduction:** Reducing the potential for human error, which can happen when files are copied manually, especially when dealing with large numbers of files.

**Scheduling and Regularity:** Once automated, the script can be scheduled to run at regular intervals (e.g., daily, weekly), ensuring that file copying is done consistently without user intervention.

**Importance**

The importance of automating file copying with a script and creating a script to copy files from one folder to another automatically lies in several key benefits that can greatly enhance efficiency and reliability in everyday workflows. Here's why it's so important:

**1. Time Savings**

Manually copying files from one folder to another can be tedious and time-consuming, especially when dealing with large volumes of files. Automating this process with a script eliminates the need for human intervention, saving significant amounts of time, especially in repetitive tasks like backups or routine file organization.

**2. Minimizing Errors**

Human error is inevitable when performing repetitive tasks, such as copying and pasting files. A script, on the other hand, can execute the task with precision, ensuring that files are copied correctly without mistakes like missing files or copying them to the wrong location.

**3. Increased Productivity**

By automating file copying, users can focus on more valuable tasks, knowing that the script will take care of the file management process. This leads to improved overall productivity as time is spent on higher-priority activities.

**4. Consistency and Reliability**

Scripts ensure that file copying is done the same way every time. This consistency is crucial in many environments (e.g., backups, data syncing) where reliability is essential to ensure that all necessary files are transferred correctly and in a timely manner.

**5. Scalability**

As the volume of files increases, manually managing file copying becomes increasingly impractical. A script can handle large amounts of data without additional effort, allowing the process to scale with the growing needs of the system.

**6. Scheduling and Automation**

With a script, you can automate file copying to occur at specific times, such as during off-hours or at regular intervals (e.g., daily backups). This means file management tasks are carried out automatically, without the need for constant attention.

**7. Cost Efficiency**

In many professional or business environments, automating repetitive tasks reduces the need for manual labor, leading to cost savings. Additionally, reduced errors and increased efficiency contribute to a better use of resources.

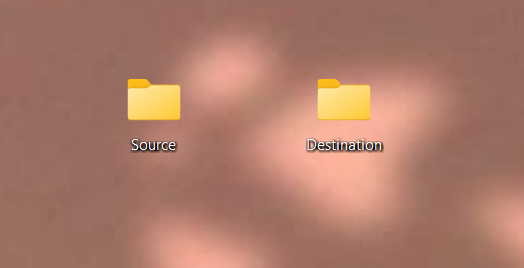
**8. Customization and Flexibility**

Scripts can be easily tailored to meet specific needs, such as excluding certain files, copying only new or modified files, or integrating the process with other workflows. This flexibility ensures that the automation aligns perfectly with your requirements.

**Step-by-Step Overview**

Step 1

Create two folders and name it as Source and Destination, and then store the files on Desktop



Step 2

Store some files inside Source Folder to automate it



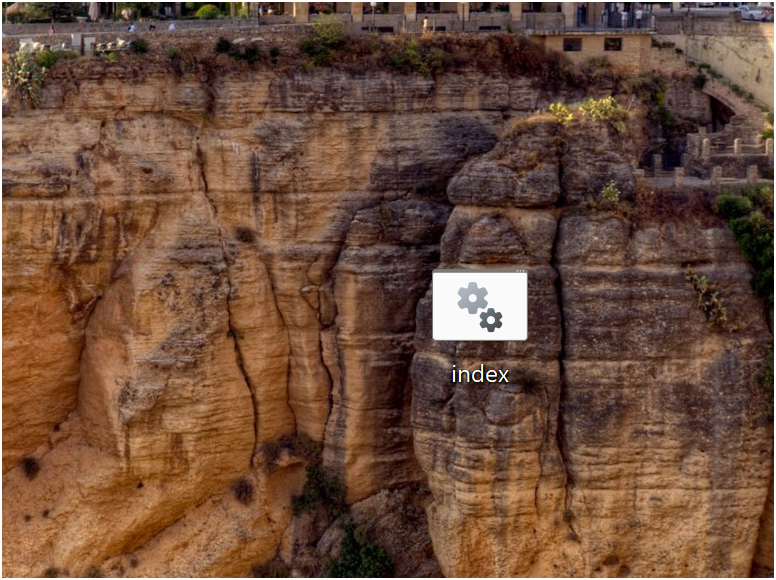
Step 3

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



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.



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.



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.





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.