

Handling Bulk AI Image Downloads & Organizing for Better Results

For Apple users: Scroll down to the bottom

1. Unpacking Multiple .zip Files at Once

When you download a large number of AI-generated images - especially from Midjourney - you'll often end up with many .zip files that need to be managed.

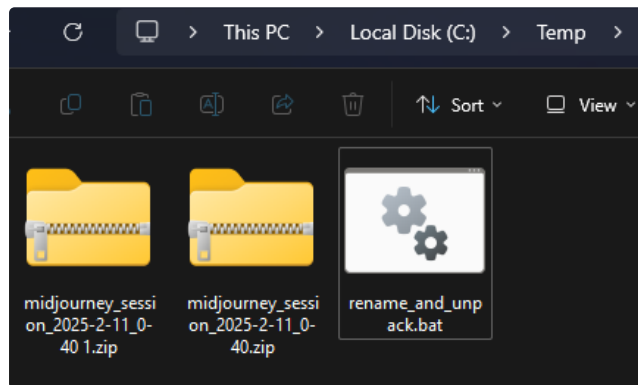
The following script automatically extracts all .zip files in the same folder in one go - this saves you a lot of time.

📎 [Windows: rename_and_unpack.bat](#) (3.37 KB)

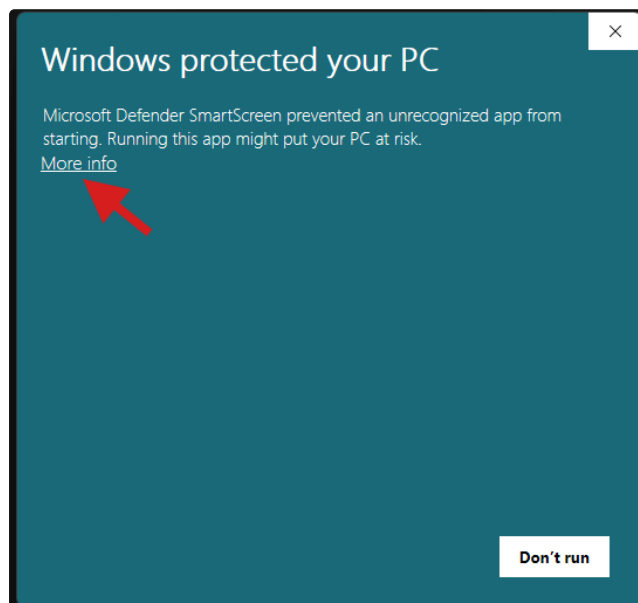
📎 [MacOS: renameandunpack.sh](#) (3.14 KB)

💡 How it works:

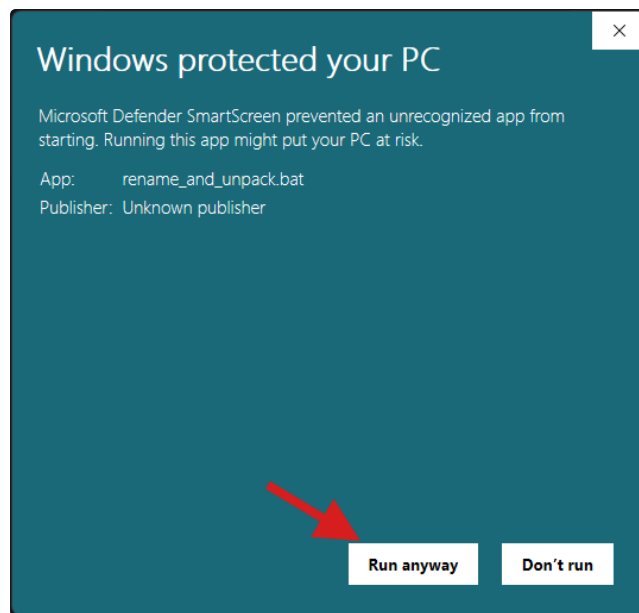
- Download the batch file "rename_and_unpack.bat"
- Copy it into the same folder where all your ZIP files are stored (e.g., C:/Temp)



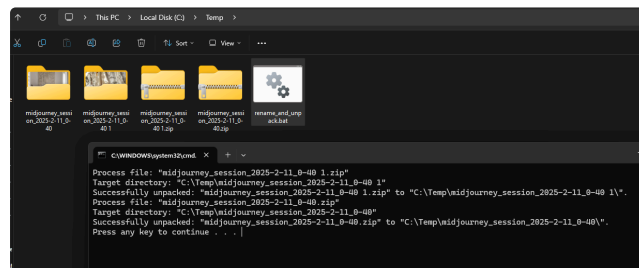
- Run the batch file. For the first time you might get a warning, click on "More info".



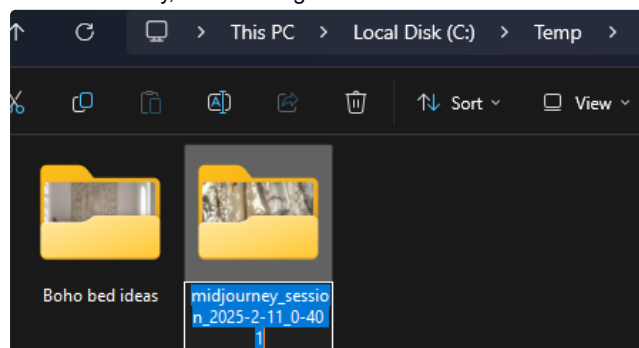
- Click on "Run anyway"



- The batch file will:
 - Rename all .ZIP files by removing special characters (to prevent errors)
 - Unpack all ZIP files automatically
- Once the process is finished, you'll see a confirmation message in the command line window.



- Clean Up:
 - You can now delete all ZIP files and the `rename_and_unpack.bat` file, because they are no longer needed.
 - If needed later, you can find them in the Recycle Bin.
- Organize Your Images:
 - If necessary, sort the images into folders named after their respective keywords from the overview sheet.



Download & Store ZIP Files:

- Collect all ZIP files in a single location.
- Tip: I recommend using `C:/Temp` for consistency, but you can choose another location if preferred.

2. Shuffle and rename files

AI-generated images usually have consecutive timestamps based on their creation order.

Since Midjourney generates four images per prompt, they often appear sequentially, making them easy to recognize as AI-generated.

To avoid this pattern, it's best to shuffle the images within the same keyword folder.

The following script automates this process:

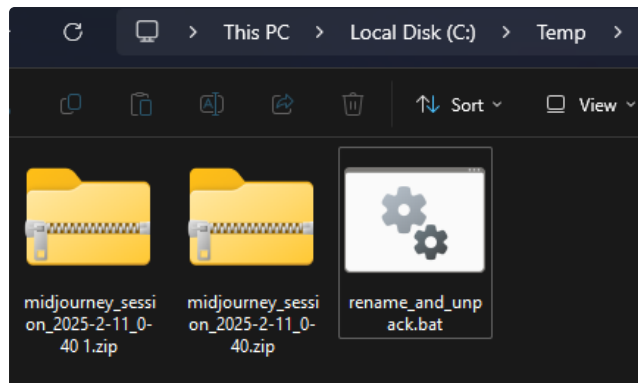
- It enters each subfolder
- Shuffles all existing images
- Renames them using the subfolder name (= keyword)
- Renumbers them to create a more natural and randomized appearance

📎 [Windows: shuffle_and_rename_files_in_subfolders.bat](#) (3.6 KB)

📎 [MacOS: shuffle_and_rename_files_in_subfolders.sh](#) (2.75 KB)

💡 How it works:

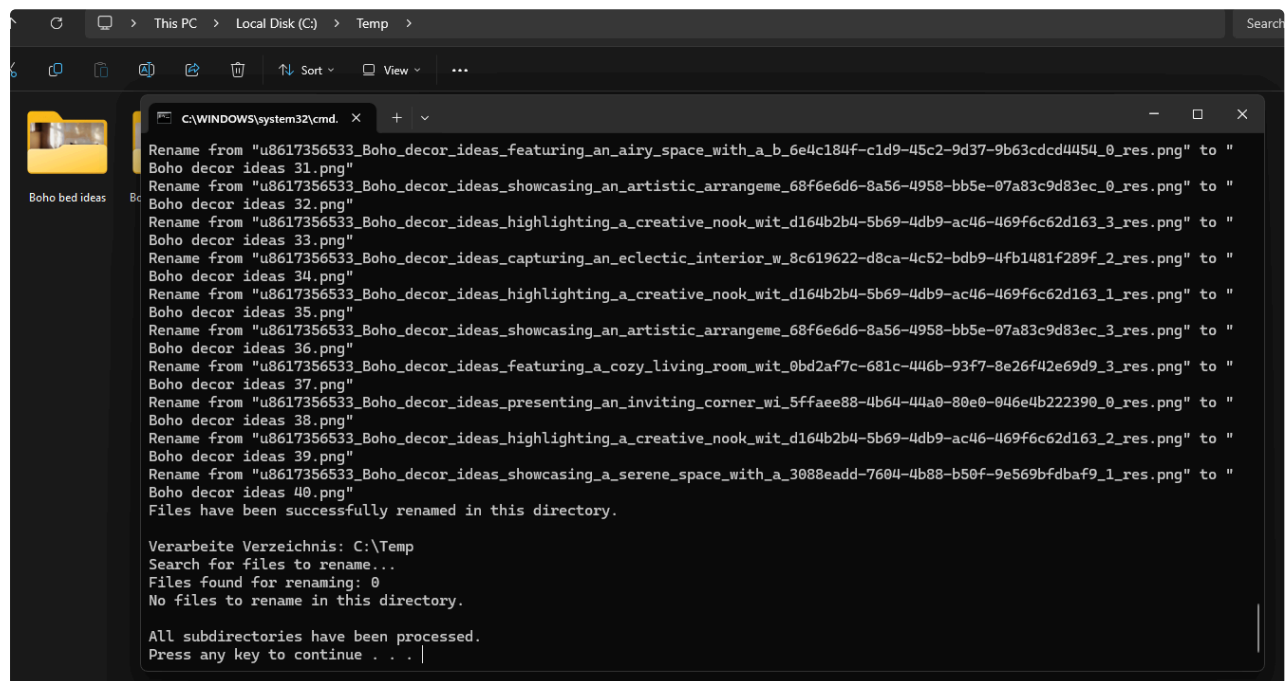
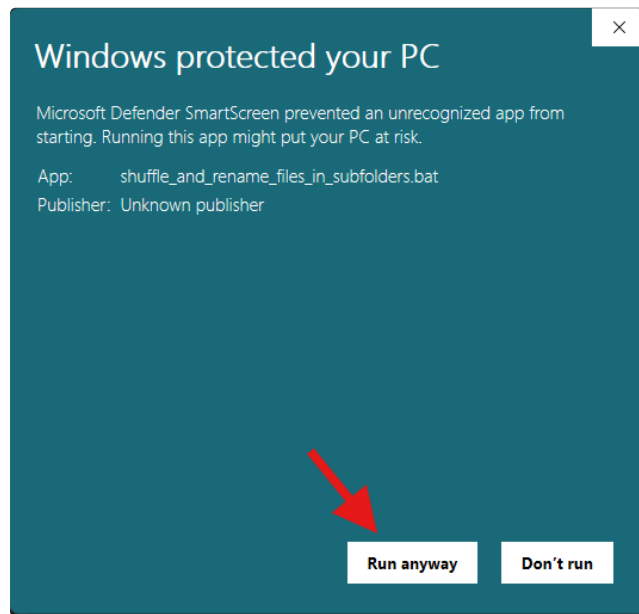
- Download the batch file "shuffle_and_rename_files_in_subfolders.bat" and place it in the same folder where all your ZIP files are stored (e.g., C:/Temp).



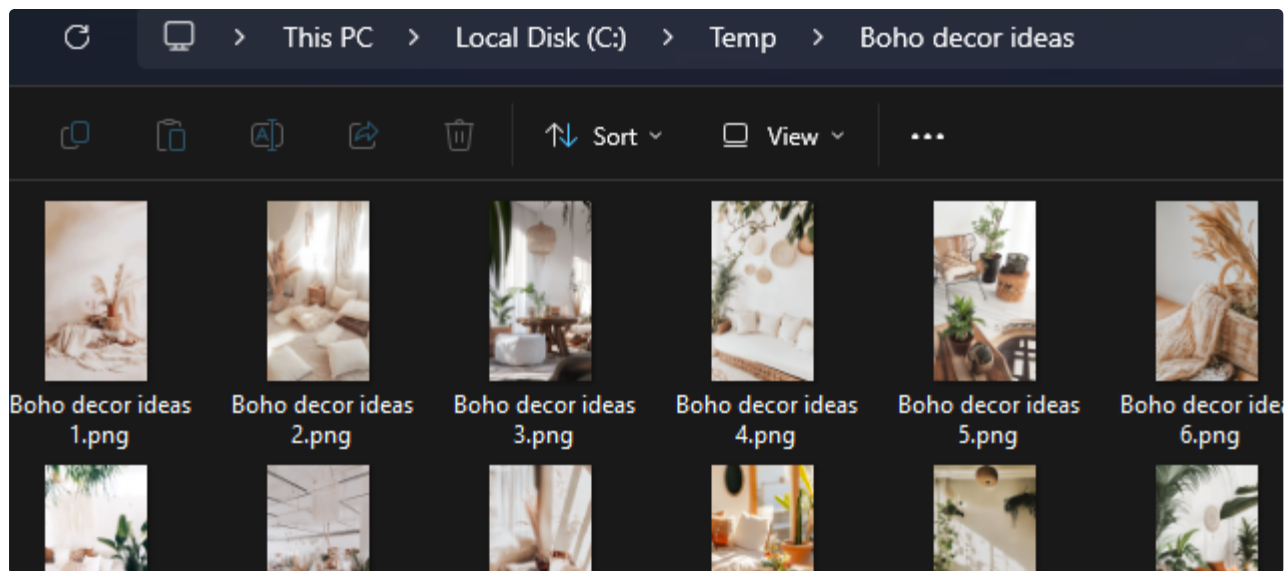
Run batch file, for the first time you might get a warning, click on "More info".



- Then click on "Run anyway"



- Now you have a fantastic result, ready for Pinterest.



For MacOS:

Depending on the authorization, the script may not be directly executable on MacOS.

But no worries, I'll show you how to create it by yourself and how to make it executable!

These adjustments should make the program work on Apple devices. Not all steps are necessary for all devices, in some cases it depends on the certain device or user settings.

This is why some of the entries are duplicated.

In this example, the script is called "renameandunpack.sh". You can of course call it whatever you like. At the point where I write "renameandunpack.sh", you should then enter your chosen name so that the program knows which script it should access.

The procedure works for both the unpacking and the shuffle and rename script.

Make sure that the programs and the images you want to access are in the same folder. For the sake of simplicity, I will call the path PATH/FOLDER.

1. Create file and insert script - depending on the setting, the program cannot be opened. To do this, we simply create the script ourselves (via "Terminal"). Enter the following:

```
~ % touch renameandunpack.sh
~ % nano renameandunpack.sh
```

-> insert the script there and ensure that no lines have slipped. Afterwards, save the script in exactly the same folder in which the images etc. are saved.

2. Make the script executable and grant access to the folders:

```
~ % chmod +x renameandunpack.sh
~ % sudo chmod -R 777 PATH/FOLDER
FOLDER % sudo chmod -R 755 PATH/FOLDER
FOLDER % ./renameandunpack.sh open -a PATH/FOLDER
FOLDER % open -a PATH/FOLDER
```

3. Make the script executable in the corresponding folder:

```
FOLDER % chmod +x renameandunpack.sh  
FOLDER % touch renameandunpack.sh  
FOLDER % nano renameandunpack.sh
```

4. Run program:

```
FOLDER % bash renameandunpack.sh
```

As already mentioned: some of the steps listed will not be absolutely necessary, but they won't do any harm either. If the script is already running with fewer steps, even better!