

Automatic File Organiser — User Guide

Why This Matters

File organisation reduces manual effort, prevents duplicates, and ensures clear separation of file types. This updated script adds **intelligent duplicate handling** (renaming on conflicts), fine-grained uncategorised handling (using the extension as folder name), and cleaner modular design. Without these, files risk overwriting, misplacement, or being skipped.

What It Is

A Python 3 script (`file_organiser.py`) that:

- Scans a target directory.
- Categorises files by type or extension.
- Moves files into corresponding folders (e.g., Documents, Images, **.xyz** for uncategorised types).
- Avoids overwriting by generating unique filenames on conflicts.
- Logs all actions and errors.
- Skips hidden/system files.

How It Works

Core Components

1. Configuration

- Defined categories:
 - Documents
→ `.pdf`, `.doc`, `.docx`, `.txt`, `.xls`, `.xlsx`, `.ppt`, `.pptx`, `.odt`, `.rtf`
 - Images → `.jpg`, `.jpeg`, `.png`, `.gif`, `.bmp`, `.tiff`, `.svg`, `.webp`
 - Videos → `.mp4`, `.avi`, `.mov`, `.wmv`, `.flv`, `.mkv`, `.webm`
 - Music → `.mp3`, `.wav`, `.aac`, `.flac`, `.ogg`, `.wma`
- Miscellaneous → placed in MISC folder.
- Uncategorised → a folder named after the extension (e.g., **.xyz** → **xyz** folder).

2. Logging

- Written to `file_organiser.log` using timestamped INFO and ERROR logs.

3. Duplicate Handling

- Checks for name conflicts; if a file `example.pdf` exists, renames the new file as `example_1.pdf`, `example_2.pdf`, etc.

4. Hidden/System File Skipping

- Skips files starting with `.` (dotfiles).

How To Use It

1. Prerequisites

- Python 3 installed (`python3 --version`).

2. Setup

- Place `file_organiser.py` anywhere.

3. Execution

- Open terminal.
- Navigate to script directory.
- Run:
`python3 file_organiser.py`

4. Provide Source Directory

- When prompted, enter the path (absolute or relative) of the directory to organise.

5. Result

- Files are sorted, renamed if necessary, and moved.
- Example structure after execution:
`/target-dir/`

- `Documents/`
- `└─ report.pdf`
- `Images/`
- `└─ photo.jpg`
- `mp4/`
- `└─ special_video.mp4`
- `MISC/`
- `└─ unnamed_file`
- `file_organiser.log`

Detailed Behaviours

Condition	Action
File matches known category extension	Moved into category folder (e.g., Documents, Images).
File has unknown extension <code>.xyz</code>	Moved into folder named <code>xyz</code> .
File has no extension	Moved into <code>MISC</code> folder.
Destination has a file with same name	Renamed using incremental suffix (e.g., <code>file_1.ext</code>).
Hidden/system file (starts with <code>.</code>)	Ignored and skipped.
Directory errors (nonexistent, not dir)	Printed and logged; script exits without crash.

What Happens If This Script Didn't Exist

- Manual sorting, risking overwrites or skipped files.
- No safety net for duplicate filenames → data loss.
- No logs → no traceability or debugging after bulk operations.

Future Enhancements (Forward-Looking)

- Add config file support for custom categories.
- Provide command-line arguments for headless automation.
- Add `.ini` or `.yaml` configuration for user-defined behaviours.
- Integrate scheduler (cron, Windows Task Scheduler) for periodic runs.
- Include detailed statistics report (how many files moved, by type).

Licence

Educational use; modify and apply at your discretion.