

02-Find-Duplicates_V01.qmd

Chapter-1-File-Organization-And-Backup

Yahya Nazer - ChatBizDB.Com

2026.06.18

Table of contents

1. Purpose	2
Key Features:	2
Safety Philosophy:	2
2. Inputs	2
System Requirements:	2
Required Input Files:	3
Processing Criteria:	3
Flexible Column Detection:	3
3. Steps	4
Step 1: Script Execution	4
Step 2: CSV File Selection	4
Step 3: Automatic Folder Creation	4
Step 4: CSV Processing and Analysis	4
Step 5: File Moving Operations	4
Step 6: Report Generation	5
4. Outputs	5
Primary Output Folder:	5
Detailed Move Report:	5
Console Output:	6
Operation Statistics:	6
Usage Guidelines	7
Pre-Processing Checklist:	7
Post-Processing Steps:	7
Recovery Procedures:	7
Error Handling	7
Common Issues and Solutions:	7
Safety Features	8
Data Protection:	8
Recovery Options:	8

Technical Notes	8
Integration with File Listing Script	9
Resources & Downloads	9
Contact Us	9

1. Purpose

The Duplicate File Mover is a safety-focused utility designed to process CSV reports containing duplicate file information and move identified duplicate files to a secure staging area. This tool provides a safe alternative to immediate deletion, allowing users to review and verify duplicates before permanent removal.

Key Features:

- **Safe file management** - Files are moved (not deleted) for easy recovery
- **Cross-platform compatibility** (Windows, macOS, Linux)
- **Intelligent CSV parsing** with automatic column detection
- **Conflict resolution** with automatic filename suffixes
- **Comprehensive error handling** and detailed logging
- **GUI-based file selection** for ease of use
- **Detailed operation reporting** with move tracking
- **Automatic staging folder creation**

Safety Philosophy:

This script prioritizes data safety by moving files to a review folder rather than deleting them immediately, providing users with a recovery path if needed.

2. Inputs

System Requirements:

- **Python 3.6+** installed on your system
- **tkinter** (usually included with Python installation)

- **Standard library modules** (no additional installations required)

Required Input Files:

The script requires **one primary input**:

CSV Report File

A CSV file containing duplicate file analysis with the following expected structure:

Column	Expected Header	Description	Required Values
A	Selector	File classification	“DUPLICATE” for files to move
C	Path	Complete file path	Valid file system paths
I	Duplicate_Index	Duplicate count	Integer > 1 for files to move

Processing Criteria:

Files are moved only when **both conditions** are met:

- Column A (Selector) = “DUPLICATE”
- Column I (Duplicate_Index) > 1

Flexible Column Detection:

The script includes intelligent column detection with fallback options:

- **Selector column:** Searches for “selector” keyword or uses first column
- **Path column:** Searches for “path” keyword or uses third column
- **Duplicate_Index column:** Searches for “duplicate_index” keyword or uses ninth column

3. Steps

Step 1: Script Execution

```
python 02-find-duplicates-v01.py
```

Step 2: CSV File Selection

- A GUI file dialog will open automatically
- Navigate to your CSV report file containing duplicate analysis
- Select the CSV file and click “Open”
- Supported formats: `.csv` files

Step 3: Automatic Folder Creation

The script automatically:

- Creates a `to_be_deleted` folder in the same directory as the script
- Prepares the staging area for moved files
- Ensures proper permissions for file operations

Step 4: CSV Processing and Analysis

The script will:

- Read and analyze the CSV file structure
- Detect column headers automatically
- Apply filtering criteria (DUPLICATE status + Index > 1)
- Display processing progress in the console

Step 5: File Moving Operations

For each qualifying file, the script:

- Verifies file existence at the specified path
- Handles filename conflicts with numeric suffixes

- Moves files safely to the `to_be_deleted` folder
- Logs all operations for reporting

Step 6: Report Generation

- Creates a detailed move report (`move_report.txt`)
- Displays operation summary in console
- Provides next steps guidance

4. Outputs

Primary Output Folder:

Location: `./to_be_deleted/` (created in script directory)

Contains all moved duplicate files with:

- **Original filenames preserved** when possible
- **Conflict resolution** using numeric suffixes (e.g., `file_1.txt`, `file_2.txt`)
- **Safe staging environment** for review before deletion

Detailed Move Report:

File: `./move_report.txt` (saved in script directory)

Report Structure:

```
Duplicate File Move Report
```

```
=====
```

```
Total files moved: [number]
```

```
Files not found: [number]
```

```
Files with errors: [number]
```

```
SUCCESSFULLY MOVED FILES:
```

```
-----  
From: [original_path]  
To:   [destination_path]
```

FILES NOT FOUND:

```
-----  
[list of missing file paths]
```

FILES WITH ERRORS:

```
-----  
File: [problematic_file_path]  
Error: [error_description]
```

Console Output:

Real-time feedback including:

- **Step-by-step progress** indicators
- **File processing status** for each operation
- **Column detection results** and mappings
- **Operation summary** with counts and statistics
- **Warning messages** for missing or problematic files
- **Next steps guidance** for post-processing

Operation Statistics:

The script tracks and reports:

- **Files successfully moved:** Count and details
- **Files not found:** Missing files from original paths
- **Files with errors:** Movement failures with error descriptions
- **Processing criteria:** Applied filters and logic

Usage Guidelines

Pre-Processing Checklist:

1. **Verify CSV format:** Ensure your duplicate report has the expected columns
2. **Check file paths:** Confirm that file paths in the CSV are current and accessible
3. **Backup important data:** Consider creating backups before running the script
4. **Review criteria:** Understand which files will be moved based on the filtering logic

Post-Processing Steps:

1. **Review moved files:** Examine contents of `to_be_deleted` folder
2. **Verify duplicates:** Confirm that moved files are indeed duplicates
3. **Check move report:** Review `move_report.txt` for any errors or warnings
4. **Final cleanup:** Delete the `to_be_deleted` folder when satisfied

Recovery Procedures:

If you need to restore moved files:

1. Navigate to the `to_be_deleted` folder
2. Locate the files you want to restore
3. Move them back to their original locations (paths listed in move report)
4. Use the move report as reference for original file locations

Error Handling

Common Issues and Solutions:

Files Not Found:

- **Cause:** File paths in CSV are outdated or files were already moved/deleted
- **Solution:** Check if files exist at specified locations before running script

Permission Errors:

- **Cause:** Insufficient permissions to move files or create folders
- **Solution:** Run script with appropriate permissions or change file ownership

CSV Format Issues:

- **Cause:** Unexpected CSV structure or encoding problems
- **Solution:** Verify CSV has expected columns and UTF-8 encoding

Filename Conflicts:

- **Cause:** Multiple files with same name being moved to destination
- **Solution:** Script automatically handles this with numeric suffixes

Safety Features

Data Protection:

- **Non-destructive operations:** Files are moved, never deleted
- **Conflict resolution:** Automatic filename suffixes prevent overwrites
- **Comprehensive logging:** All operations tracked for audit trail
- **Error isolation:** Individual file errors don't stop batch processing

Recovery Options:

- **Move report reference:** Complete mapping of original to new locations
- **Staging folder:** All moved files accessible in single location
- **Reversible operations:** Easy to restore files to original locations

Technical Notes

- **Memory efficient:** Processes files individually without loading entire dataset
- **UTF-8 encoding:** Supports international characters in file paths
- **Cross-platform paths:** Handles Windows, macOS, and Linux path formats
- **Automatic delimiter detection:** Works with comma, tab, and other CSV separators

- **Robust error handling:** Graceful handling of permission and access issues

Integration with File Listing Script

This script is designed to work seamlessly with the File Listing Script output:

1. Use File Listing Script to generate comprehensive file reports
2. Use duplicate detection tools to analyze and mark duplicates in CSV
3. Use this Duplicate File Mover to safely stage duplicate files for review
4. Manually review and confirm before final deletion

Resources & Downloads

Download all Python scripts and resources from this book:

www.chatbizdb.com/python-book

All code examples, templates, and additional materials are available for download. Please do not share this link - Thank you.

Copyright © 2025 ChatBizDB.com. All rights reserved.

No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law.

For permission requests, write to the publisher at: [Email us](#)

Contact Us

If you have any question or comment please contact us at [Email us](#) and make sure you add **python-book** at the beginning of the subject.

```
[1] "Time taken to run the Quarto document: 0.0963501930236816 seconds"
```