How to choose a duplicate finder app for Mac: A transparent evaluation methodology

Objective criteria for comparing Duplicate Finder apps.

This methodology is based on hands-on experience in developing and testing macOS utilities (Nektony, since 2011).

Table of Contents

TL;DR

For users

A safe duplicate finder app must:

- Ask for confirmation before removal
- Move files to Trash (not delete permanently)

- Warn or restrict deleting the last file in a duplicate group
- Never delete system files

> To check any app quickly, use the calculator

For reviewers and QA

The methodology includes 53 criteria across 3 levels:

- L1 (8 criteria) — blockers; failing any criterion means the app is "not recommended"
- L2 (23) — important for usability
- L3 (22) — bonuses

Why this methodology exists

In the Mac App Store and on developer websites, there are dozens of duplicate finder apps.
All of them promise to "quickly find and safely remove duplicates."
But how to find out which one is actually safe?

A wrong choice can be costly:

- Important files might be deleted (false positives)
- Photo Library or Music Library might be damaged
- Real duplicates might be missed (low accuracy)
- The app crashes on large file sets

The Nektony team has been developing macOS utilities since 2011. Over the years, we've analyzed the behavior of dozens of competing products and collected feedback from hundreds of thousands of users. Based on that experience, we created a transparent evaluation methodology: **53 criteria that do matter for safe operation**.

We publish it transparently so that:

- You can evaluate any app on your own
- Journalists and reviewers have an objective comparison framework

- Quality standards in the duplicate removal category become higher for everyone

Who this methodology is for

Audience	How to use
Regular users	Use the calculator — it walks you through key criteria
Journalists and reviewers	Download the full methodology to compare apps objectively
QA engineers	The PDF includes step-by-step tests, scripts, and a report template
Developers	Use it as a requirements checklist for your own product

Terms and definitions

Before starting the evaluation, it's important to understand the terminology:

Term	Meaning
Exact Duplicate	Files with identical content (bit-for-bit)
Similar File	Files with visual or audio similarity (not byte-for-byte)
Duplicate Group	A set of 2+ files considered to be copies of each other
Original	The file that remains after duplicates are removed
False Positive	A unique file incorrectly marked as a duplicate
False Negative	A real duplicate that was not detected

Metrics and result calculation

For objective evaluation purposes, we use standard metrics from Information Retrieval:

Core metrics

Metric	Abbreviated	Description
True Positive	TP	Duplicate Groups identified correctly
False Positive	FP	Unique files incorrectly marked as duplicates
False Negative	FN	Real duplicates that were not found

How we count TP/FP/FN (unit of measurement)

- The unit of measurement is a Duplicate Group (a set of 2+ files).
- TP: An expected group is counted as found if the app found all files in the group and combined them into one group.
- FP: The number of groups that contain at least one unique file presented by the app as a duplicate (not an exact duplicate matched by content).
- FN: The number of Duplicate Groups the app failed to find, or found incompletely (at least one file missing).

If the app otherwise groups files (merging/splitting)

- Splitting one expected group into several counts as FN (group found incompletely).
- Merging two expected groups into one counts as FN for both groups (incorrect grouping).
- Passing L1 requires full and correct grouping of Exact Duplicates.

Calculated metrics

Metric	Formula	What it measures	Target
Recall	TP / (TP + FN)	What portion of real duplicates were found?	≥ 99%
Precision	TP / (TP + FP)	What portion of found items are truly duplicates?	100%

Measurement scope

- Recall is measured on datasets 1–4 (known number of duplicates).
- Precision (FP=0) is checked on datasets 1–6, where dataset 6 contains edge cases and traps.
- Results are averaged over 2–3 runs to eliminate random deviations.

Example calculation

```
Dataset 3 (10,000 photos):
— Expected Duplicate Groups: 5,000
— Groups found: 4,980
— False positives: 0

Recall = 4980 / 5000 = 99.6% ✅
Precision = 4980 / (4980 + 0) = 100% ✅
```

Core principle: safety is higher priority than speed

When evaluating apps, we follow a simple priority:

Safety > Accuracy > Usability > Speed

A fast app that deletes important files is useless.
A slow but safe app is useful.

Scoring system: 3 levels of criteria

All 53 criteria are divided into three levels by importance:

Level	Name	Criteria	Weight	Max points
L1	Must Have	8	×3	24

Level	Name	Criteria	Weight	Max points
L2	Should Have	23	×2	46
L3	Nice to Have	22	×1	22
	Total	**53**		**92**

How to interpret the score

Score	Recommendation
71-92	✅ Recommended
46-70	⚠️ With caveats
< 46	❌ Not recommended

Important! Failing any Level 1 criterion automatically falls under "Not recommended," regardless of the total score.

> Quick check: Use the interactive calculator — it walks you through the criteria and calculates the score automatically.

Evaluation criteria

Level 1: Must Have (blocking criteria)

Data safety criteria.
Failing any of them means a real risk of file loss.

The app must correctly find duplicates, protect against accidental removal, and guarantee recoverability.

If at least one criterion is not met, the app is **Not recommended,** regardless of the total score.

#	Criterion	Description
1.1	Recall ≥ 99%	The app must find almost all real duplicates
1.2	False positives = 0%	Unique files must never be marked as duplicates
1.3	Removal confirmation	Deletion only after explicit user confirmation
1.4	Last file protection	The app must prevent deleting ALL files in a group without special confirmation — at least one must remain
1.5	File integrity	Remaining originals must not be damaged
1.6	System file protection	Must not search or show duplicates in ~/Library/
1.7	Stability	No crashes on test sets (up to 200,000 files)
1.8	Running without Full Disk Access	Must run in Documents, Desktop, Downloads, etc. without Full Disk Access

RED FLAGS 🚩

If an app does any of the following, don't use it.

Safety

- *Allows deleting ALL files in a group without warning*
- *Deletes files immediately without confirmation*
- *Does not show where files are deleted to (Trash or permanently)*
- *Crashes when scanning large numbers of files*

Monetization transparency

Not always dangerous, but often a sign of poor UX and wasted time.

- *Scans for free but requires payment to delete (paywall trap)*
- *Price is not visible until checkout*
- *Free version limitations are not explained upfront*
- *Intrusive ads interfere with app usage*

Level 2: Should Have

Functionality and usability criteria.

They cover search settings, performance, working with special sources (Photo Library, iCloud, external drives), and the quality of the results interface. Without these features, an app can still be used, but with limitations in certain scenarios.

#	Criterion	Description
2.1	Minimum file size	Option to skip small files
2.2	Search types	Files, folders, similar photos, similar audio
2.3	Folder duplicates	Option to find duplicate folders
2.4	Skip list	Option to toggle off folders, files, and extensions
2.5	Baseline time	Scanning speed is reasonable
2.6	Pause/cancel	Option to stop the scan
2.7	Progress indicator	Shows remaining scan progress
2.8	External/network drives	USB, NAS, SMB
2.9	Photo Library	Works with Photos.app
2.10	Hidden files	Finds dotfiles and files flagged as hidden
2.11	Quick Look	File preview before removal
2.12	Path/size/date	Full information for each file
2.13	Why duplicate	Shows hash or confirms "bit-for-bit" match
2.14	Space to be freed	Shows how much disk space will be freed
2.15	Autoselect	Automatic file selection
2.16	Autoselect rules	Configurable rules for automatic selection

#	Criterion	Description
2.17	Open in Finder	Option to open a file/folder in Finder
2.18	Removal preview	List of files before removal
2.19	Trash vs Permanent	Choice between moving to Trash or deleting permanently
2.20	Restore deleted files	Deleted files must go to Trash and be recoverable
2.21	Error handling	Clear error messages and option to keep going
2.22	Help on error	Suggestions on how to fix issues when errors occur
2.23	Partial results	Shows what succeeded and what failed if an error occurs

Level 3: Nice to Have

Extra features for advanced users.

Accessibility (VoiceOver, localization), advanced result handling (sorting, search, export), and alternative actions (moving files to a chosen folder instead of deleting them, replacing duplicates with symlinks). The lack of these features is not critical, but their availability makes the app more usable in certain use cases.

#	Criterion	Description
3.1	Dark mode	Supports the macOS system Dark Mode
3.2	VoiceOver	Accessible for visually impaired users
3.3	Localization	Supports 5+ interface languages
3.4	Onboarding	First-launch tutorial
3.5	Help/support	Built-in help, FAQ, and support contacts
3.6	Clear warnings	Warnings explain the risk and recommended action
3.7	Side-by-side comparison	Compare two files next to each other

#	Criterion	Description
3.8	Sorting/grouping	Sort results by size, date, or path
3.9	Search in results	Search and filter field within results
3.10	Metadata (EXIF)	Displays EXIF data for photos
3.11	Deselect All	Clear all selected files with one click
3.12	Move instead of Delete	Move files to a folder instead of deleting them
3.13	Replace with symlink	Replace a duplicate with a symbolic link
3.14	Folder merge	Merge contents of duplicate folders
3.15	Export report	Export results to CSV or PDF
3.16	Session saving	Save results after quitting the app
3.17	Operation log	View history of removed duplicate files
3.18	Add to Skip List	Add exclusions directly from results
3.19	Processing 200K+ files	Stable performance with very large volumes
3.20	Interim results	Shows results before the scan is finished
3.21	APFS clones	Finds clones and warns that removal won't free space
3.22	iCloud	Works with evicted (cloud-only) files

How we test: 7 datasets

For objective evaluation purposes, we use standardized test datasets:

Dataset	Contents	What we test
1	Duplicates and trap files (230 groups × 3 copies)	Detection accuracy
2	200 large files (50 groups × 4 copies)	Performance on large files
3	10,000 photos (5,000 groups × 2 copies)	Typical user scenario
4	200,000 files (1,000 groups × 200 copies)	Stress test, stability
5	1000 files (100 groups × 10 copies, 15 nesting levels)	Deep folder structures
6	Edge cases*	Hidden files, APFS clones, symlinks, .app bundles, unicode names, long paths
7	1 file in 8 locations	Access to different user folders

Datasets used for metric calculation

- Recall (L1.1) is calculated on datasets DS1–DS4, where the expected number of duplicate groups is known in advance.
- FP=0 (L1.2) is checked on datasets DS1–DS5.
- Datasets DS6–DS7 are used for functional checks (location access and permissions) and are not included in Recall or FP metric calculation.

***What is included in Edge cases (Dataset 6)**

Non-standard macOS cases, where many apps crash:

- Zero-byte files
- Files with extended attributes (xattr)
- Symlinks and hardlinks
- App bundles must be treated as single files, not folders

- Hidden files (by name `.file` and by hidden flag)
- APFS clones
- Files with Unicode characters in names (emoji, hieroglyphs)
- Files with very long paths (>255 characters)

Performance baseline

For SSD and Apple Silicon:

Dataset	Excellent	Acceptable	Unacceptable
2 (large files)	< 30 sec	< 90 sec	> 180 sec
3 (10K photos)	< 60 sec	< 180 sec	> 300 sec
4 (200K files)	< 5 min	< 15 min	> 30 min

For Intel Mac, ×1.5-2 of these values is acceptable.

RAM:

- < 1 GB — excellent
- < 2 GB — acceptable
- < 4 GB — borderline
- > 4 GB or crash — unacceptable

Risks and common pitfalls

> **Read before testing!**

Dangerous removal scenarios

Scenario	Risk
Photo Library	Photos.app Direct removal may damage Photos.app library
Music Library	Same for Music.app
System files (~Library/, /System/)	Risk of breaking macOS
Files in use	May be locked or damaged

False results

Case	Problem
APFS clones	Technically duplicates, but removal does NOT free space
iCloud evicted	File may auto-download during scan
Resource forks / xattr	Files may differ in invisible metadata
Sandbox limitations	App Store version sees fewer files

Common testing mistakes

- Compare App Store and non-App Store versions without taking into account Sandbox
- Test without Full Disk Access (incomplete results)
- Run only one scan instead of 2–3 (scan times vary)
- Not to check file recovery BEFORE bulk removal

Testing procedure

Below is a step-by-step guide for full application testing. Each step is tied to specific criteria.

Preparation

Step 0.1. Set up the test environment

1. Create a new macOS user or use a fresh session. We recommend running the test on the latest macOS on a physical device. Do not use a virtual machine, it distorts the results.
2. Make a note of the configuration:
   - macOS version: _______________
   - Hardware (CPU, RAM, disk type): _______________
   - Test date: _______________
3. Quit all unnecessary applications.
4. Open Activity Monitor to track RAM usage.

Step 0.2. Prepare the application

1. Make a note of app details:
   - App name and version: _______________
   - Source (App Store / website): _______________
   - Price / monetization model: _______________
2. Install the app
3. Launch it and grant Full Disk Access (System Settings → Privacy & Security → Full Disk Access)
4. Check support:
   - Apple Silicon support: Native / Rosetta / unknown
   - Recent app update: _______________

Step 0.3. Create test datasets

All test datasets will be placed in an isolated folder: `~/DuplicateTest/`:

```
~/DuplicateTest/
├── Dataset_01_AccuracyTest/    # 851 files, 230×3 duplicates, the rest are t
├── Dataset_02_LargeFiles/      # 50 files × 1–30 MB
├── Dataset_03_Photos/          # 10,000 photos
├── Dataset_04_StressTest/      # 200,000 files
├── Dataset_05_DeepNesting/     # 1000 files, 15 levels
├── Dataset_06_EdgeCases/       # Non-standard cases
└── logs/
```

📦 Scripts for creating test datasets: Download archive

The archive contains bash scripts to automatically create all 7 datasets.
Download and run the scripts in Terminal, they will generate the test datasets.

Dataset 1 — Accuracy Test:

- 851 files: 690 duplicates (230 groups × 3) + 161 traps
- Traps: similar names, different sizes, different dates, different metadata
- Generates manifest.json for automated result validation
- Expected result: 230 duplicate groups, 3 files per group, 0 false positives

Dataset 2 — Large files:

- 200 files: 50 originals × 4 copies (originals + copy_A + copy_B + copy_C)
- File size: 1–30 MB (dmg, mp4, zip, mov, pkg)
- Dataset size: ~3.1 GB
- Expected result: 50 groups of 4 files

Dataset 3 — Photos:

- 10,000 files: 5,000 groups of 2 files
- Folder A: 2,500 photos + 2,500 APFS clones (cp -c)
- Folder B: 2,500 photos + 2,500 regular copies in subfolder C
- File size: ~500 KB (JPEG imitation)
- Expected result: 5,000 groups of 2 files

Dataset 4 — Stress test:

- 200,000 files: 1,000 unique × 200 copies
- File size: 20 KB (txt, dat, bin, tmp, log)
- Dataset size: 4 GB
- Expected result: 1,000 groups of 200 files

Dataset 5 — Deep nesting:

- 1,000 files: 100 unique × 10 copies
- Nesting: up to 15 levels, including hidden folders (.hidden, .cache)
- File size: ~50 KB (txt, pdf, jpg, png, docx, mp4, zip, json, xml, html)
- Expected result: 100 groups of 10 files

Dataset 6 — Edge Cases:

- 19 files: hidden (by name and flag), symlinks, hardlinks
- 2 .app bundles, files with extended attributes (xattr)
- 6 zero-byte files
- Expected result: 4 duplicate groups (16 files in groups)

Dataset 7 — Location test:
One file (100 MB) in 8 folders:

- `~/Documents/dup_location_test/`
- `~/Desktop/dup_location_test/`
- `~/Downloads/dup_location_test/`
- `~/Movies/dup_location_test/`
- `~/Music/dup_location_test/`
- `~/Pictures/dup_location_test/`
- `~/Library/dup_location_test/` (requires Full Disk Access)
- `~/DuplicateTest/`

Expected result: 1 group of 7 files (duplicates in the system folder must be ignored)

Alternative: minimal dataset for quick testing

A ready-to-use minimal dataset for quick app validation. It's enough to check L1 criteria and basic L2 criteria.

📦 Download minimal dataset: QuickTest.zip (~150 MB)

Dataset structure:

```
~/DuplicateTest/
├── originals/                    ← 10 original files of different types
│   ├── photo.jpg                 ← photo with EXIF
│   ├── document.pdf
│   ├── video.mp4
│   ├── music.mp3
│   ├── archive.zip
│   ├── text.txt
│   ├── .hidden1.txt              ← hidden file #1
│   ├── .hidden2.txt              ← hidden file #2
│   ├── large_file.dmg            ← file >100 MB
│   ├── emoji_🎉.txt              ← unicode in the name
│   └── similar_name_A.txt        ← TRAP (unique content)
│
├── copies/                       ← copies of all files from originals
│   ├── photo.jpg
│   ├── ... (all files)
│   └── similar_name_B.txt        ← TRAP: similar name, DIFFERENT content!
│
├── nested/                       ← deep nesting test
│   └── level1/
│       └── level2/
│           └── level3/
│               └── photo.jpg     ← third copy photo.jpg
│
├── .hidden_folder_A/             ← hidden folder #1
│   ├── file1.txt
│   └── file2.txt
│
├── .hidden_folder_B/             ← hidden folder #2 (duplicate)
│   ├── file1.txt
│   └── file2.txt
│
├── folder_dup_A/                 ← regular folder #1
│   ├── doc1.txt
│   ├── doc2.txt
│   └── doc3.txt
│
└── folder_dup_B/                ← regular folder #2 (duplicate folder_dup_
    ├── doc1.txt
    ├── doc2.txt
    └── doc3.txt
```

Expected result:

What should be found	Groups	Files per group	Folders per group
video.mov (originals + copies + nested)	1	3	0
photo.jpg (originals + copies + nested)	1	3	0
8 files from originals/copies (2 copies each)	8	2	0
Files in hidden folders (file1.txt, file2.txt)	2	2	0
Files in regular folders (doc1-3.txt) or folder_dup_A and folder_dup_B	2 1	3 all files in the folder	0 2
.hidden_folder_A and .hidden_folder_B	1	all files in the folder	2
Total duplicates	**15**	**32**	**4**

What should NOT be found:

- `similar_name_A.txt` and `similar_name_B.txt` — different files, NOT duplicates (trap)

This dataset checks:

- ✅ Detection accuracy (1.1)
- ✅ False positives (1.2) — similar_name trap

- ✅ Hidden files (2.10) — .hidden1.txt, .hidden2.txt
- ✅ Hidden folders — .hidden_folder_A/B
- ✅ Folder duplicates (2.3) — folder_dup_A/B
- ✅ Deep nesting — nested/level1/level2/level3
- ✅ Unicode in names — emoji_🎉.txt
- ✅ Large files — large_file.dmg

⚠️ **Limitations:** The minimal dataset does not allow you to test stability on large volumes (1.7) or precise performance baseline (2.5). For full testing, use the scripts.

Phase 0. Initial app review

Goal: Prepare the app for testing and evaluate transparency.

0.1. Monetization and transparency

Launch the app and answer:

Question	Answer
Are free version limitations clear BEFORE you start?	Yes / No
Is the price visible before purchase?	Yes / No
No "trap": free scan but paid removal with no warning?	Yes / No
Ads do not interfere with app usage?	Yes / No / No ads

Red flag: If the app shows results but requires payment to delete, that's a paywall trap.

0.2. Interface navigation

1. Launch the app and complete onboarding (if any).
2. Find and make a note of:

- Where folders for scanning are selected: ________________
- Where settings are (filters, skip list): ________________
- Where scanning starts: ________________
- Where results and removal are shown: ________________

Phase 1. Checking Level 1 (Must Have) criteria

Goal: Verify critical safety requirements. Failing any criterion = "Not recommended".

Before getting started, scan the folder ~/DuplicateTest/ (datasets 1–6).

These scan results will be used in tests below.

1.1. Recall ≥ 99%

Why it matters: The app must find all real duplicates. Missing duplicates means the user won't free the expected disk space.

How to check:

1. Scan the folder `~/DuplicateTest/` entirely.
2. Make a note of the number of groups and duplicates found.
3. Compare it with the expected numbers.
4. Calculate: Recall = Found / Expected × 100%

Summary table:

Dataset	Expected groups	Found	Recall
1	230		
2	50		
3	5000		
4	1000		
5	100		

Result:

- ✅ Recall ≥ 99% on all datasets
- ❌ FAIL: Recall < 99%

1.2. False positives = 0%

Why it matters: Unique files must never be marked as duplicates. A mistake here can lead to loss of important data.

How to check:

1. Open 10 random groups in the scan results.
2. For each group, make sure that the files are truly identical:
   - Same size
   - Same content (open and compare)
3. Pay special attention to Dataset 1 (traps) and Dataset 6 (edge cases).
4. Make a note of the number of false positives found: ________

Result:

- ✅ False Positives = 0
- ❌ FAIL: Unique files found in duplicate groups

1.3. Removal confirmation

Why it matters: An accidental click must not cause data loss. There must be a confirmation step.

How to check:

1. Select 3–5 files to be deleted.
2. Click Delete/Remove.
3. Is there a confirmation dialog?
4. Does it show:
   - Number of files
   - Space size to be freed

    - Where files will be deleted (Trash / permanently)

Result:

- ✅ The app shows confirmation with details
- ❌ FAIL: The app deletes files immediately without confirmation

1.4. Last file protection

Why it matters: You must not be able to delete all files in a group — at least one must remain. Otherwise, all of the data will be lost.

How to check:

1. Find a group of 2 duplicates.
2. Try selecting BOTH files for removal.
3. Click Delete/Remove.
4. What happens?

Result:

- ✅ The app does not allow you to select both files
- ✅ The app allows you to select, but shows a warning before removal
- ❌ FAIL: The app allows you to remove both files without warning

1.5. File integrity

Why it matters: After deleting a duplicate, the remaining original must not be damaged.

How to check:

1. After deleting a duplicate, locate the remaining original.
2. Open it with the appropriate app:
    - Photo → Preview or Photos
    - Video → QuickTime
    - Document → relevant app
3. Does it open fully and correctly?
4. Repeat for 3-5 various file types.

Result:

- ✅ All files open correctly
- ❌ FAIL: At least one file is damaged or won't open

1.6. System file protection

Why it matters: The app must not suggest deleting system files from ~/Library/.

How to check:

1. Scan the `~/Library/` folder.
2. Check whether the app shows files from there in results.
3. If so, try selecting a file from `~/Library/Preferences/` for removal.

Result:

- ✅ The app does not scan ~/Library/ or does not show files from there in results
- ✅ The app shows them but blocks removal and explains the reason
- ⚠️ The app shows and warns, but allows removal
- ❌ FAIL: The app allows you to delete system files without warning

1.7. Stability

Why it matters: The app must not crash, even on large datasets.

How to check:

1. Scan Dataset 4 (200,000 files).
2. During the scan, check:
   - If there are any crashes?
   - If there are any freezes (beachballs)?
3. In Activity Monitor, check:
   - RAM usage: ________ MB
   - If memory grows uncontrollably?

Result:

- ✅ The app is stable and completes the scan, RAM < 4 GB
- ❌ FAIL: Crash, freeze or RAM > 4 GB

1.8. Running without Full Disk Access

Why it matters: The app must run for users who don't grant Full Disk Access.

How to check:

1. Disable Full Disk Access for the app: System Settings → Privacy & Security → Full Disk Access → toggle off
2. Restart the app.
3. Try scanning `~/Documents/`, `~/Desktop/`, `~/Downloads/`.
4. See how the app behaves.

Result:

- ✅ The app allows you to select a folder and scan. Standard user folders can be scanned.
- ✅ There is a clear message if FDA is required for protected folders.
- ❌ FAIL: Crash, unclear error, or the app does not run at all.

Level 1 summary

#	Criterion	Result	Note
1.1	Recall ≥ 99%	✅ / ❌	Recall: ___%
1.2	False positives = 0%	✅ / ❌	FP: ___
1.3	Removal confirmation	✅ / ❌	
1.4	Last file protection	✅ / ❌	
1.5	File integrity	✅ / ❌	
1.6	System file protection	✅ / ⚠️ / ❌	
1.7	Stability	✅ / ❌	RAM: ___ MB

#	Criterion	Result	Note
1.8	Running without Full Disk Access	✅ / ❌	
	Total L1:	**___ / 8**	

Is there at least one ❌ ? → The app automatically gets the status "**Not recommended.**"

Phase 2. Checking Level 2 (Should Have) criteria

Goal: Check the app functionality and usability.

Search settings

2.1. Minimum file size

Why it matters: The app lets you skip tiny files and focus on large duplicates that actually take up space.

How to check: Is there an option in settings to set a minimum file size?

Result:

- ✅ Yes, a specific value can be set
- ❌ No such setting

2.2. Search type selection

Why it matters: A user may only need one type of search — for example, file duplicates only, without similar photos. Choice saves time.

How to check: Can you choose what to search for?

- File duplicates
- Similar photos
- Similar audio

Result:

- ✅ Yes, multiple types to choose from

- ❌ Only one search type

2.3. Folder duplicates

Why it matters: Sometimes entire folders get duplicated (for example, project backups). Removing a whole folder is more efficient than file by file.

How to check: Is there a mode to find duplicate folders (not just files)?

Result:

- ✅ Yes, there is such mode
- ❌ No, files only

2.4. Skip List

Why it matters: It lets you skip folders you don't want scanned (node_modules, .git, caches) and speeds the search.

How to check: Can you skip folders, files, or extensions from the search?

Result:

- ✅ Yes (folders + files + extensions)
- ⚠️ In part (only one of these)
- ❌ No

Performance

2.5. Baseline time

Why it matters: If scanning is too slow, the app isn't practical for regular use.

How to check: Measure scan time and compare with the baseline:

Dataset	Excellent	Acceptable	Unacceptable
2 (large files)	< 30 sec	< 90 sec	> 180 sec
3 (10K photos)	< 60 sec	< 180 sec	> 300 sec

Dataset	Excellent	Acceptable	Unacceptable
4 (200K files)	< 5 min	< 15 min	> 30 min

Result:

- ✅ Time is within "Excellent" or "Acceptable"
- ❌ Time falls under "Unacceptable"

2.6. Cancel/pause

Why it matters: Users need to control the process. If a scan runs too long, there must be a way to stop it.

How to check:

1. Start scanning a large folder (Dataset 4).
2. Are there Pause and Cancel buttons?
3. Do they work?

Result:

- ✅ Pause + cancel, both work
- ⚠️ Cancel only
- ❌ No way to stop

2.7. Progress indicator

Why it matters: Without a progress indicator, it's impossible to tell if the app is working or frozen. It's crucial for scanning large drives and folders.

How to check: Do you see progress during the scan?

- Progress bar
- Percentage complete
- Number of processed files
- Number of duplicates found

Result:

- ✅ Informative progress (2+ elements)
- ❌ No indicator or useless one (stuck at 99%)

Special sources

2.8. External/network drives

Why it matters: Duplicates often pile up on external drives and NAS. The app should handle different storage locations.

How to check:

1. Connect a USB drive and create a small test folder with duplicates.
2. Does the app show it as a source?
3. Does scanning work?
4. (If so) Check NAS/SMB.

Result:

- ✅ USB + NAS work
- ⚠️ USB only
- ❌ Not supported

2.9. Photo Library

Why it matters: The Photos.app library is one of the largest duplicate sources for regular users. The app is supposed to handle this library.

How to check:

1. Select `~/Pictures/Photos Library.photoslibrary` as a source.
2. Run the scan.
3. Does it find duplicates inside the library?

Result:

- ✅ It scans and shows duplicates from Photo Library
- ⚠️ It shows the library as a file but does not show its photos as duplicates
- ❌ Not supported

2.10. Hidden files

Why it matters: Hidden files (dotfiles or flagged as hidden) can also be duplicates. Skipping them means an incomplete scan.

How to check:

1. Use Dataset 6 (Edge Cases) — hidden files are in there.
2. Or create `.hidden_test.txt` and a copy elsewhere.
3. Scan — are hidden duplicates found by the app?

Result:

- ✅ The app finds hidden files
- ❌ The app does not find them

Results interface

2.11. Quick Look

Why it matters: Quick Look is the standard macOS preview. Its availability lets you quickly review files before removal.

How to check: Select a file in results, press Space — does Quick Look work?

Result:

- ✅ Yes
- ❌ No

2.12. Path/size/date

Why it matters: This info helps you decide which file to keep based on directory, size, or modification date.

How to check: For each file in results, is the following shown:

- Full path
- File size
- Date modified (or date created)

Result:

- ✅ All info available (3/3)
- ⚠️ In part (1-2 out of 3)
- ❌ No info

2.13. Why duplicate

Why it matters: Algorithm transparency builds trust. Users should understand why files are considered duplicates.

How to check: Do results show why files are duplicates?

- Hash is shown (MD5, SHA)
- The "Bit-for-bit identical" message is shown
- Match by size + content is shown

Result:

- ✅ Reason shown
- ❌ Not shown

2.14. Space to be freed

Why it matters: The whole point of searching duplicates is to free up space. Users should see how much space they'll actually free up.

How to check: Can you see how much space will be freed after deleting selected files?

Result:

- ✅ Yes, I can see (for group or selected files)
- ❌ No, I can't see

2.15. Autoselect

Why it matters: With thousands of duplicates, manual selection isn't realistic. Autoselect saves time and reduces mistakes.

How to check: Is there an automatic selection feature?

Result:

- ✅ Yes
- ❌ No

2.16. Autoselect rules

Why it matters: Different users have different priorities: keep the newest file, keep files in a specific folder, etc.

How to check: Can you configure autoselect rules?

- Keep newest
- Keep oldest
- Keep by path (e.g., ~/Documents)
- Keep least nested

Result:

- ✅ There are configurable rules (2+)
- ⚠️ There is only one option
- ❌ No configuration (or no autoselect)

2.17. Open in Finder

Why it matters: Sometimes you need to see the file in its folder context — what files are next to it and what the folder's name is.

How to check: Can you open the file or folder in Finder directly from the results?

Result:

- ✅ Yes (right click or button)
- ❌ No

2.18. Removal review

Why it matters: It's the last chance to review what will be deleted. It is especially important to review selected files after using autoselect.

How to check: Can you review the full list of selected files before removal?

Result:

- ✅ Yes
- ❌ No

2.19. Trash vs Permanent

Why it matters: Trash is a safety net against accidental deletions. But sometimes you really do need to delete files permanently (for example, sensitive data). It's best when you have a choice.

How to check: Is there an option to either move files to Trash or delete them permanently?

Result:

- ✅ Both options are available
- ❌ Only one option is available

2.20. Restore works

Why it matters: If a file is deleted by mistake, there must be a way to recover it. The Trash provides the standard macOS recovery mechanism for that.

How to check:

1. Delete 2-3 files through the app.
2. Open Trash — are the files there?
3. Restore the file (right click → Put Back).
4. Does the file return to its original folder and open correctly?
5. Or, is there a restore command inside the app? (even better, you can immediately see "your" deleted files)

Result:

- ✅ Files are in Trash, restore works
- ❌ Files are permanently deleted or cannot be restored

2.21. Error handling

Why it matters: Errors are inevitable (locked files, no access permissions). The app should handle them gracefully without interrupting the entire process.

How to check:

1. Lock one duplicate file: Get Info → Locked ✓.
2. Try deleting this file through the app.
3. What happens?

Result:

- ✅ Clear error message, removal of other files continues
- ❌ Crash, unclear error, or the entire process stops

2.22. Help on error

Why it matters: A good app not only reports an error but also suggests how to fix it.

How to check: When an error occurs, does the app suggest how to fix it?

Result:

- ✅ Yes, there is a suggestion (for example, "unlock the file")
- ❌ No, only an error message

2.23. Partial results

Why it matters: After an operation, the user should clearly understand what actually happened — how many files were removed, how many were not, and why.

How to check: If removal is partial (some files removed successfully, some failed) — does the app show what was removed and what wasn't?

Result:

- ✅ Yes, a detailed report
- ❌ No information

Level 2 Summary

#	Criterion	Result
2.1	Minimum file size	✅ / ❌
2.2	Search type selection	✅ / ❌

#	Criterion	Result
2.3	Folder duplicates	✅ / ❌
2.4	Skip List	✅ / ⚠️ / ❌
2.5	Baseline time	✅ / ❌
2.6	Cancel/pause	✅ / ⚠️ / ❌
2.7	Progress indicator	✅ / ❌
2.8	External/network drives	✅ / ⚠️ / ❌
2.9	Photo Library	✅ / ⚠️ / ❌
2.10	Hidden files	✅ / ❌
2.11	Quick Look	✅ / ❌
2.12	Path/size/date	✅ / ⚠️ / ❌
2.13	Why duplicate	✅ / ❌
2.14	Space to be freed	✅ / ❌
2.15	Autoselect	✅ / ❌
2.16	Autoselect rules	✅ / ⚠️ / ❌
2.17	Open in Finder	✅ / ❌
2.18	Removal preview	✅ / ❌
2.19	Trash vs Permanent	✅ / ❌
2.20	Restore works	✅ / ❌
2.21	Error handling	✅ / ❌
2.22	Help on error	✅ / ❌

#	Criterion	Result
2.23	Partial results	✅ / ❌
	Total L2:	**___ / 23**

Note: ⚠️ = 0.5 points

Phase 3. Checking Level 3 (Nice to Have) criteria

Goal: Check additional features. Their lack is not critical, but having them improves the overall experience.

For each criterion: feature present and working → ✅, otherwise → ❌

3.1. Dark Mode

Why it matters: Many macOS users prefer Dark Mode. A native app should automatically adapt to system appearance settings.

How to check: System Settings → Appearance → Dark. Does the app switch to Dark Mode?

Result:

- ✅ Yes
- ❌ No

3.2. VoiceOver

Why it matters: Visually impaired users should also be able to use the app. VoiceOver support is a sign of a well-built native app.

How to check: Enable VoiceOver (Cmd+F5) and try navigating. Are key elements read out loud?

Result:

- ✅ Yes, navigation works
- ❌ No or in part

3.3. Localization

Why it matters: Not all users speak English. Localization expands the audience and improves UX for non-English speakers.

How to check: Are languages other than English supported?

Supported languages: ________________

Result:

- ✅ 5+ languages
- ⚠️ 2-4 languages
- ❌ English only

3.4. Onboarding

Why it matters: New users may not understand how the app works. Onboarding lowers the learning curve.

How to check: Is there a first-launch tutorial?

Result:

- ✅ Yes
- ❌ No

3.5. Help/support

Why it matters: If something is unclear or goes wrong, the user should be able to find help easily.

How to check: Is there built-in help, FAQ, or quick access to support (Contact Us)?

Result:

- ✅ Yes
- ❌ No

3.6. Clear warnings

Why it matters: A good warning explains the risk and helps the user make the right decision.

How to check: Do warnings explain:

- What will happen (action)
- Why it matters (risk)
- What to do (recommendation)

Result:

- ✅ Yes, warnings are helpful (2+ elements)
- ❌ No, just "Are you sure?"

3.7. Side-by-side comparison

Why it matters: For visual content (photos, documents), it's convenient to compare files next to each other to pick the best one.

How to check: Can you compare two files from a group side by side in the interface?

Result:

- ✅ Yes
- ❌ No

3.8. Sorting/grouping

Why it matters: Sorting by size helps remove the largest duplicates first and free up space faster.

How to check: Can you sort results by size, date, or path? Group them?

Result:

- ✅ Yes
- ❌ No

3.9. Search in results

Why it matters: With thousands of results, you need a way to quickly find a specific file or folder.

How to check: Is there a search or filter field in the results?

Result:

- ✅ Yes
- ❌ No

3.10. Metadata (EXIF)

Why it matters: EXIF helps you understand which photo is more "original" — by capture date, camera, geolocation.

Use real camera photos with EXIF data.

How to check: Does the app display EXIF data (capture date, camera, geolocation) for these photos?

Result:

- ✅ Yes
- ❌ No

3.11. Deselect All

Why it matters: If autoselect picked the wrong files, you should be able to quickly reset and start over.

How to check: Is there a one-click option to clear all selections?

Result:

- ✅ Yes
- ❌ No

3.12. Move instead of Delete

Why it matters: Sometimes you may want to move duplicates to a separate folder for manual review instead of deleting them right away.

How to check: Is there an option to move files to a folder instead of deleting them?

Result:

- ✅ Yes
- ❌ No

3.13. Replace with symlink

Why it matters: A symlink lets you keep a file "in place" (for compatibility), but free up space — the link itself takes up almost no space.

How to check: Can you replace a duplicate with a symbolic link?

Result:

- ✅ Yes
- ❌ No

3.14. Folder merge

Why it matters: If two folders duplicate each other in part, it's more convenient to merge them than delete files one by one.

How to check: Is there a feature to merge contents of duplicate folders?

Result:

- ✅ Yes
- ❌ No

3.15. Export report

Why it matters: For documentation, auditing, or analysis in other tools, exporting results is useful.

How to check: Can you export results (CSV, PDF, txt)?

Format(s): _______________

Result:

- ✅ Yes
- ❌ No

3.16. Session saving

Why it matters: Scanning can take time. If you close the app, results shouldn't be lost.

How to check: Close the app after scanning, reopen it. Are the results saved?

Result:

- ✅ Yes
- ❌ No

3.17. Operation log

Why it matters: A history of actions helps you understand what was deleted and recover files if needed.

How to check: Is there a history of removed files or an activity log?

Where to find: _______________

Result:

- ✅ Yes
- ❌ No

3.18. Add to Skip List

Why it matters: If you see a file in the results that shouldn't be removed, it's convenient to skip it right from there.

How to check: Can you add a file or folder to the skip list directly from the results?

Result:

- ✅ Yes
- ❌ No

3.19. Processing 200K+ files

Why it matters: Some users have hundreds of thousands of files. The app should handle large volumes without freezing.

How to check: On Dataset 4 (200,000 files), is the app stable and the UI responsive?

Result:

- ✅ Yes
- ❌ No (lags, freezes)

3.20. Interim results

Why it matters: During long scans, it's convenient to see interim results and start working with them before the scan finishes.

How to check: During scanning, are already found duplicates displayed before completion?

Result:

- ✅ Yes
- ❌ No

3.21. APFS clones

Why it matters: APFS clones use Copy-on-Write. In this case, deleting a clone does NOT free up space. The user needs to know that.

How to check:

1. Create a file clone: select a file → Cmd+D (creates an APFS clone).
2. Scan the folder with the original and the clone.
3. Does the app warn that deleting the clone will NOT free up space?

Result:

- ✅ Recognizes clones + warns
- ⚠️ Finds them, but does not warn
- ❌ Does not recognize clones

3.22. iCloud

Why it matters: Many users store files in iCloud. The app should properly handle files that exist "in the cloud only".

How to check:

1. Put a file into iCloud Drive.
2. Right click → Remove Download (file remains only in the cloud).
3. Scan iCloud Drive — does the app see the evicted file?

Result:

- ✅ The app works with evicted files
- ❌ The app does not see them or downloads them automatically

Level 3 Summary

#	Criterion	Result
3.1	Dark Mode	✅ / ❌
3.2	VoiceOver	✅ / ❌
3.3	Localization	✅ / ⚠️ / ❌
3.4	Onboarding	✅ / ❌
3.5	Help/support	✅ / ❌
3.6	Clear warnings	✅ / ❌
3.7	Side-by-side comparison	✅ / ❌
3.8	Sorting/grouping	✅ / ❌
3.9	Search in results	✅ / ❌
3.10	Metadata (EXIF)	✅ / ❌
3.11	Deselect All	✅ / ❌
3.12	Move instead of Delete	✅ / ❌
3.13	Replace with symlink	✅ / ❌
3.14	Folder merge	✅ / ❌
3.15	Export report	✅ / ❌
3.16	Session saving	✅ / ❌
3.17	Operation log	✅ / ❌

#	Criterion	Result
3.18	Add to Skip List	✅ / ❌
3.19	Processing 200K+ files	✅ / ❌
3.20	Interim results	✅ / ❌
3.21	APFS clones	✅ / ⚠️ / ❌
3.22	iCloud	✅ / ❌
	Total L3:	**___ / 22**

Note: ⚠️ = 0.5 points

Total score calculation

Formula

```
Total score = (L1 × 3) + (L2 × 2) + (L3 × 1)
Example: (7 × 3) + (18 × 2) + (15 × 1) = 21 + 36 + 15 = 72 балла
```

Your result

```
Level 1: ___ / 8 × 3 = ___
Level 2: ___ / 23 × 2 = ___
Level 3: ___ / 22 × 1 = ___
─────────────────────────────
TOTAL:                ___ / 92
```

Interpretation

Score	Recommendation
71-92	✅ Recommended
46-70	⚠️ With caveats
< 46 or L1 failure	❌ Not recommended

Final report template

App details

String	Value
App name	
Version	
Source (App Store / website)	
Price / monetization model	
Test date	
macOS version	
Hardware (CPU, RAM, disk)	
Tester	

Performance

Dataset	Time	RAM (MB)	Groups found	Expected	Recall
1				230	
2				50	

Dataset	Time	RAM (MB)	Groups found	Expected	Recall
3				5000	
4				1000	
5				100	
6	—	—		4	
7	—	—		1	

Total score

```
Level 1: ___ / 8 × 3 = ___
Level 2: ___ / 23 × 2 = ___
Level 3: ___ / 22 × 1 = ___
─────────────────────────────
TOTAL:                ___ / 92
```

Conclusion

Blockers (L1 failure):

____

Main drawbacks:

____

Strengths:

____

RECOMMENDATION:

- ✅ Recommended (71-92, no L1 failures)
- ⚠️ With caveats (46-70)
- ❌ Not recommended (<46 or L1 failure)

Evaluate the app yourself

We've created an interactive calculator to help you evaluate any app based on our methodology:

→ Open the evaluation calculator

The calculator will:

- Walk you through 53 criteria
- Calculate the score automatically
- Show the recommendation
- Save progress (you can come back later)
- Export the result to a file

Download the full methodology

For QA engineers and journalists, the full version of the methodology with test dataset creation scripts is available:

📦 **Download test dataset scripts**

📦 **Download minimal dataset QuickTest.zip**

QuickTest.zip (~150 MB) — for quick validation of L1 and basic L2 criteria

Includes:

- Detailed description of all 53 criteria
- Scripts for generating test datasets
- Final report template
- Minimal dataset for quick testing

Frequently asked questions

Why do you publish the methodology if you develop a Duplicate Finder yourself?

We believe transparency builds trust. A transparent methodology allows anyone to verify our statements. If our product is good, it will show. If not, we'll learn what needs to be improved.

Can this methodology be used for other categories of apps?

In part. The removal safety criteria (L1) and UX criteria (L3) apply to any cleaning utilities. The specific criteria (Photo Library, APFS clones) apply only to duplicate finders.

Can I suggest a new criterion?

Yes! Email us at support@nektony.com. We'll consider your suggestion for the next version.

Disclaimer

This methodology is provided "as is" for reference only.

Limitations

- Test results may vary depending on the app version, macOS version, hardware configuration, and test data.
- This methodology is not exhaustive and does not cover all possible use cases.
- Scores and recommendations are advisory in nature.

Liability

- Nektony shall not be liable for decisions made based on test results using this methodology.
- The user shall be solely liable for choosing their software and for the consequences of using them.
- It is always recommended to create backups before deleting files.

Conflict of interests

Nektony develops Duplicate File Finder — an app in the tested category. We publish this methodology transparently and apply it to our own products. Independent verification of results is welcome.

License

Questions and feedback

Found an issue in the methodology? Want to suggest a new criterion?

Contact us: support@nektony.com

Last updated: February 2026 (version 1.0)

Related materials

- Duplicate Finder Evaluation Calculator
- Comparison of Popular Duplicate Finder Apps for Mac (2025)

- Duplicate File Finder by Nektony