

DDiff GUI

Directory Compare & One-Way Synchronization

User Manual

Version 1.0

Free Pascal / Lazarus edition

Author: T. Chieng

Document date: September 2026

Contents

1. Introduction.....	4
What DDiff does.....	4
2. System Requirements.....	4
3. Installation and Launch.....	4
4. The DDiff Window.....	5
Row colours in the difference list.....	5
The Action column.....	5
5. Quick Start.....	6
6. Comparing Directories.....	6
How each path is classified.....	6
Comparing by content (MD5).....	7
Resolving a pair by hand.....	7
7. Synchronizing.....	8
Path checks and the mass-deletion guard.....	8
Reusing the comparison.....	9
Parallel copying.....	9
Per-file progress bars.....	9
8. Options Reference.....	9
9. The Mirror Preset.....	10
10. Delayed Deletion.....	10
11. Exclusions.....	11
12. Profiles — Save and Load.....	11
13. Network Credentials for UNC Paths.....	11
Entering credentials.....	12
How the stored password is protected.....	12
14. Command-line (Headless) Mode.....	12
Command-line options.....	12
Exit codes.....	13
15. Logs, Reports, and CSV Export.....	13
16. Statistics.....	13
17. Stopping a Run.....	14
18. Troubleshooting.....	14
19. Important Cautions.....	14

1. Introduction

DDiff GUI compares two directory trees and, on request, synchronizes them in one direction — from a source (the “left” side) to a destination (the “right” side). It works much like Robocopy, with one important addition: it can delay deletions on the destination instead of removing files immediately, giving you a grace period to recover anything purged by mistake.

The tool is designed for repeatable backup and mirroring jobs. Every run produces a full, timestamped log and a per-run report, and any job can be saved to a profile and recalled later exactly as configured.

What DDiff does

1. **Scans** the source tree and the destination tree.
2. **Classifies** every relative path as New, Newer, Older, Same, or Right-only.
3. **Copies** New and updated files from source to destination and — when the matching options are enabled — overwrites older files, creates empty source directories, verifies each copy, and purges files that exist only on the destination.

NOTE DDiff synchronizes in one direction only. The source is never modified; only the destination is changed.

2. System Requirements

- Windows (the tool is a native Windows application).
- Read access to the source folder, and read / write / delete rights on the destination and the log folder.
- For network locations, either your Windows session already has access, or you supply credentials (see Section 13) — a mapped drive is not required.
- No additional runtime or components are required; the program is a single self-contained executable.

3. Installation and Launch

DDiff GUI ships as a single executable, DdiffGUI.exe. There is nothing to install — copy it to any folder and run it. On first launch the window opens ready for input, with the log folder defaulting to C:\Temp\DDiff_Logs.

TIP Keep the executable and your saved profiles (.ddp files) together in one folder so a job and the tool that runs it stay in the same place.

4. The DDiff Window

The window is divided into four areas:

Input area (top). The source and destination paths, the log folder, the exclusion list, the Options group, and the Credentials and profile buttons.

Action buttons (top right). Compare, Synchronize, Stop, and the Mirror preset.

Difference list (center). A colour-coded grid of every difference found — one row per file or folder, showing its side, status, size, the action taken, the full path, and a Progress column. Click a column header to sort by that column; click the same header again to reverse the order (the # and Size columns sort numerically). An arrow marks the sorted column. While a file is being copied — during a Synchronize, or from a right-click Copy Left to Right / Right to Left — its row shows a live progress bar with the percentage in the Progress column.

Log and progress (bottom). A live log of every action, a progress bar, the current-status line, and the Export / Open buttons.

Row colours in the difference list

Colour	Meaning
Green	New file to be copied, or a completed copy
Teal	Newer on the source (will be copied)
Olive	Older on the source (copied only if Overwrite is on)
Gray	Identical on both sides (unchanged)
Maroon	Exists only on the destination (a purge candidate)
Purple	Timestamp could not be read (Unknown)
Red (bold)	An operation failed — see the log

The Action column

The Action column shows what a Synchronize would do to each entry, using the options currently selected. This is a preview in Compare mode and the starting state of each row in Synchronize mode:

Action	Meaning
to copy	A source file that will be copied (New or Newer, or Older when Overwrite is on), or a new source-only folder when Copy empty directories is on.
to purge	A destination-only entry that will be deleted, shown only when Purge destination files not in source is on.
none	Nothing will happen to this entry — for example identical files, or a destination-only entry when Purge is off.

During a Synchronize each row is updated in place as its action finishes, changing to copied, overwritten, purged, or held (for delayed deletion), or to a failure such as COPY FAILED or PURGE FAILED. The same row is reused throughout — there are no separate before-and-after rows.

5. Quick Start

1. **Set the source.** Type or browse to the source (left) folder.
2. **Set the destination.** Type or browse to the destination (right) folder.
3. **Choose options.** Tick the options you need (see Section 8), or click Mirror preset for a full mirror.
4. **Compare first.** Click Compare to see exactly what would change. Nothing is modified.
5. **Synchronize.** Click Synchronize, read the confirmation dialog, and confirm to carry out the copy and any purge.

CAUTION Always Compare before you Synchronize. Synchronize can overwrite and delete files on the destination.

6. Comparing Directories

Compare is read-only. It scans both trees, matches them by relative path, and lists every difference in the grid and in a report file. Nothing on disk is changed, so it is always safe to run.

The two trees are scanned at the same time: a helper thread walks the destination while the main thread walks the source. When the source and destination are on different drives or network shares this roughly halves the scan time, because neither side waits for the other. When both sides share the same physical disk the speed-up is smaller, since the two scans compete for one device.

How each path is classified

Status	Meaning
New	Exists only on the source
Newer	Exists on both; the source copy is newer, or a different size
Older	Exists on both; the source copy is older
Same	Identical date and size on both sides
Right-only	Exists only on the destination (a purge candidate)
Unknown	Timestamp could not be read (only with “Ignore date/time read errors”)

NOTE Comparison is case-insensitive, so File.txt and file.txt are treated as the same file — the correct

behaviour on Windows.

Comparing by content (MD5)

By default DDiff decides whether a file that exists on both sides has changed from its name, date/time and size. Tick Compare by MD5 checksum (in Options) to compare by content instead: for each file present on both sides, the two copies' MD5 hashes are compared. If they differ the file is marked Newer and is always copied left-to-right, regardless of the Overwrite option; if they match it is Same and skipped.

Files whose sizes already differ are treated as changed without hashing (a different size always means different content), and new or right-only files are still decided by name. This mode is the reliable choice when timestamps cannot be trusted — for example when files were restored or copied by another tool and kept their old content but a new date, or the reverse. It is slower, because matched files of equal size are read in full on both sides to hash them; the status line shows a running count, each compared pair appears in the list as soon as it is checked (rather than all at once at the end), and Stop interrupts it between files.

For a large matched file (256 KB or more) the two sides are hashed at the same time on two threads — the source and destination are read concurrently — which roughly halves the wait when the two trees are on independent storage, such as a local disk and a network share. Small files are hashed sequentially, and when both sides are on the same physical disk the two reads share one device, so the parallel read gives little benefit.

TIP Compare by MD5 checksum pairs naturally with Verify copies (MD5): the first decides what to copy by content, the second confirms each copy arrived intact.

Resolving a pair by hand

After a Compare, right-click a differing file pair — a row whose status is Newer, Older, or Unknown, meaning the file exists on both sides but the contents differ — to handle it individually. A menu offers:

Menu item	Action
Compare / View	Open the pair in an external diff tool (WinDiff, Beyond Compare, ...) to inspect the differences.
Copy Left → Right	Copy the file left-to-right now, and remove the pair from the sync queue.
Copy Right → Left	Copy the file right-to-left now (overwriting the source), and remove the pair from the sync queue.
Skip (exclude from sync)	Leave both files as they are and remove the pair from the sync queue.
Configure compare tool...	Pick the external diff program (and optional extra arguments); it is remembered for next time.

The two copy actions ask for a short confirmation (they overwrite a real file), show a live progress bar in the Progress column while the file copies, then mark the row and drop that pair from the pending Synchronize — so you can resolve a few conflicts by hand and let Synchronize handle the rest. These exclusions apply to the reused comparison and are cleared if you run Compare again.

NOTE The external tool is launched as `tool.exe [extra args] "left file" "right file"`, which suits WinDiff, Beyond Compare (BCompare.exe), and most diff tools. Set it once via Configure compare tool...; it is saved in a per-user settings file.

7. Synchronizing

Synchronize performs the one-way copy from source to destination. New and Newer files are always copied. Depending on the options you set, DDiff also overwrites Older files, creates empty source directories, verifies each copy, and purges destination-only entries.

Clicking Synchronize first shows a confirmation dialog that spells out precisely what will be overwritten, purged, or held for delayed deletion. The job does not start until you confirm.

WARNING DDiff does not check the destination for free space, and with Purge on it can delete up to 100% of the destination. Double-check both paths and make sure the destination is backed up before you confirm.

Path checks and the mass-deletion guard

Before and during a run DDiff guards against common mistakes:

- **Missing source (Compare or Synchronize)** — if the source folder does not exist, an error is shown and nothing happens.
- **Missing destination (Compare)** — Compare is read-only, so a missing destination is an error; nothing happens.
- **Missing destination (Synchronize)** — DDiff asks whether you really want to create it, and proceeds only if you say yes.
- **Mass-deletion guard (Synchronize with Purge)** — if a purge would delete more than half of the files on the destination, DDiff warns you with the exact count and percentage and asks for confirmation before doing anything. Say no and nothing is changed.

NOTE In headless (command-line) mode there is no one to answer, so a missing destination is created and the mass-deletion warning is logged and the run proceeds — a scheduled job runs unattended. Both events are always written to the log.

Reusing the comparison

To avoid scanning and hashing the folders twice, Synchronize reuses the results of your most recent Compare when the source, destination, and comparison options (Compare by MD5, Ignore date/time, Ignore read errors, Exclude) are unchanged — it applies that file list directly instead of re-scanning. This is especially valuable with Compare by MD5 checksum, which would otherwise re-hash every matched file a second time. The confirmation dialog states which path it will take.

If no matching comparison is available — you changed a path or a comparison option, or clicked Synchronize without comparing first — Synchronize scans and compares on its own. Because a synchronize changes the destination, the comparison is refreshed on the next run rather than reused again; click Compare at any time to force a fresh comparison.

Parallel copying

When files are copied, DDiff uses up to four worker threads at once, so large jobs with many files finish faster; MD5 verification, when turned on, is shared across those threads too. All destination folders are created first, on a single thread, so the parallel copies never collide while creating the same folder. Progress, logging, and the difference list work exactly as before — rows simply appear in the order the copies finish rather than strictly by path.

Per-file progress bars

Each file is copied with the Windows CopyFileEx API, which reports how many bytes have been transferred as the copy runs. DDiff turns that into a live progress bar shown in the Progress column of the file's difference row, together with the percentage complete. This is most useful for large single-file copies, where you can watch the bar fill instead of waiting with no feedback. Pressing Stop cancels the copy that is in flight; the partially written destination file is then removed so no half-copied file is left behind.

8. Options Reference

Option	Effect
Overwrite newer files on destination	Copy the source file over a newer destination file. Without this, Older files are left untouched.
Purge destination files not in source	Delete (or hold) files and folders that exist only on the destination, and remove the empty directories left behind.
Copy empty directories	Recreate source-only (empty) folders on the destination.
Verify copies (MD5)	After each copy, compare the source and destination MD5 checksums and flag any mismatch.
Compare by MD5 checksum	Compare files present on both sides by content (MD5), not date/size; a differing file is marked Newer and always copied (see Section 6).
Ignore date/time (name +	Compare by name and size only. Useful across file systems with

Option	Effect
size)	different time resolutions.
Ignore date/time read errors	Treat an unreadable timestamp as Unknown instead of stopping.
Show identical files in the list	Include Same files in the grid and report (off by default, to reduce noise).
Delayed deletion / days	Instead of deleting purged files at once, move them into a holding folder and remove them only after the set number of days (see Section 10).

9. The Mirror Preset

The Mirror preset button ticks three options in one click — Overwrite newer files, Purge destination files not in source, and Copy empty directories. Together these make the destination an exact mirror of the source: every change is copied, and anything not present on the source is removed from the destination.

Mirror preset only sets the checkboxes; it does not start a run. Review the options, Compare, then Synchronize as usual.

10. Delayed Deletion

With Delayed deletion enabled, a purge does not delete destination-only files outright. Instead, DDiff moves each one — preserving its sub-folder structure — into a hidden holding folder beneath the destination:

```
<destination>\DDiff_DelayedDeletion\
```

Each moved file is stamped with the current time. On every later run, any held file older than the configured number of days is permanently removed, and emptied holding sub-folders are cleaned up. This gives you a grace period during which a mistakenly purged file can still be recovered.

NOTE The holding folder is cleaned up only while Delayed deletion is ticked, and it is always skipped by the scan — so with the option off, files already in it are left untouched, neither purged nor aged out.

TIP Prefer Delayed deletion the first few times you run a new mirror job. If a path was wrong, the “purged” files are still sitting in the holding folder.

11. Exclusions

The Exclude box takes a list of file or folder names separated by semicolons. Any entry whose name matches is skipped on both sides during the scan. For example:

```
Thumbs.db;~$temp;node_modules;.git
```

Matching is by name only (not full path) and is case-insensitive. The tool also always skips \$RECYCLE.BIN, System Volume Information, and its own DDiff_DelayedDeletion holding folder.

12. Profiles — Save and Load

A profile stores everything needed to recall a synchronization job exactly. Click Save profile... to write the current configuration to a .ddp file, and Load profile... to restore it.

A profile captures:

- the source and destination paths;
- the log folder;
- the exclusion list;
- every option, including Compare by MD5 checksum and Delayed deletion with its day count; and
- the network credentials (password stored encrypted).

Because a .ddp is a plain-text INI file, it can be kept alongside a job, copied between machines, or hand-edited. A file that is not a DDiff profile is detected and reported rather than applied. A typical profile looks like this:

```
[DDiff Profile]
Version=1
Source=D:\Data\Projects
Destination=\\backup01\share\Projects
LogFolder=C:\Temp\DDiff_Logs
Exclude=Thumbs.db;~$*.tmp
Overwrite=1
Purge=1
CopyEmptyDirs=1
Verify=0
CompareMD5=1
DelayDeletions=1
DelayDays=7
```

13. Network Credentials for UNC Paths

DDiff can read from and write to a network share by its UNC path (\\server\share\...) without mapping a drive letter. If your current Windows session already has access, nothing extra is needed. When a share needs a different login, click Credentials... and enter a username and password for the source and/or destination.

Entering credentials

- Usernames may be given as *user*, *DOMAIN\user*, or *user@domain*.
- Credentials apply only to UNC paths; they are ignored for local drives.
- Leave a section blank to use your current Windows session for that side.

Before each run DDiff authenticates every `\\server\share` with a deviceless connection (no drive letter) and disconnects afterwards. If the server is already connected under a different account, Windows does not allow a second login, so DDiff logs a warning and uses the existing connection — run `net use \\server\share /delete` first if you must switch accounts.

How the stored password is protected

When you save a profile, the password is written in an encrypted form — a random salt, a keystream derived from a built-in key, and an integrity check, shown as `SourcePassword=enc:...`. The plaintext never appears in the file.

CAUTION This protects the password from casual viewing, but it is obfuscation, not strong encryption: the key is inside the program, which keeps profiles portable between machines but means anyone with the executable could recover it. Treat a .ddp that holds a password as a secret — restrict its file permissions and prefer a dedicated low-privilege account.

14. Command-line (Headless) Mode

DDiff can run without the window. When you start it with a profile path as an argument, it loads that profile, performs the job, prints a summary, and exits with a status code — no GUI is shown. This is intended for scheduled tasks, batch files, and other automation.

```
DdiffGUI.exe "C:\Jobs\nightly.ddp"
```

Create the profile once in the window with Save profile... (Section 12), then point the command line at that .ddp file.

Command-line options

Option	Effect
/q /quiet	Suppress the per-file output on screen; print the summary only
/c /compare	Compare only (read-only; make no changes)
/s /sync	Synchronize — the default when no mode is given
/? --help	Show usage and exit

Each flag also accepts a leading dash (for example, `-q`). Logging to the profile's log folder is always on; `/quiet` only affects what is echoed to the screen — the full per-file detail is still written to the log file. The summary is always printed, even in quiet mode.

Exit codes

Code	Meaning
0	Success — no errors
1	Usage error, or the profile could not be loaded / validated
2	Completed, but with copy or verify errors (or stopped early)
3	Could not run (for example, too many copy failures)

A batch file can test the exit code and capture the console output to its own file. For a nightly backup, schedule a line such as:

```
DdiffGUI.exe "C:\Jobs\nightly.ddp" /q >> "C:\Jobs\nightly.out" 2>&1
```

TIP Use the Windows Task Scheduler to run a profile automatically. Point it at your .ddp with /q so only the summary is recorded, and check the exit code (0 means a clean run).

15. Logs, Reports, and CSV Export

Logging is always on. Two kinds of output are written to the Log folder (default C:\Temp\DDiff_Logs, editable in the window):

File	Contents
DDiff_YYYYMMDD.log	Appended, one line per action — every copy, overwrite, purge, hold, verify, and error, with a run header and a summary.
DDiff_compare_*.txt	A per-run comparison report: the full difference list plus totals.
DDiff_sync_*.txt	A per-run synchronization report: what was copied, purged, or held, plus totals.

When a profile is loaded or in use, the log file is named after that profile instead of DDiff — loading Backup_Sync.ddp writes to Backup_Sync_YYYYMMDD.log, and a headless run always uses its profile's name. Compare / Synchronize runs without a profile keep the DDiff_YYYYMMDD.log name. (The per-run report keeps its DDiff_compare_* / DDiff_sync_* name.)

The buttons along the bottom of the window let you export the grid to a CSV file, open the most recent report, or open the log folder in Windows Explorer.

16. Statistics

Every run reports counts of:

- New, Newer, Older, Same, and Right-only entries; and

- for a synchronize: files copied, verify failures, files purged (or held for delayed deletion), directories removed, total bytes copied, and the average transfer rate in MB/s.

Totals appear in the status bar, the finish dialog, the log, and the report.

17. Stopping a Run

All work runs on a background thread, so the window stays responsive. Click Stop at any time to halt a scan, comparison, or synchronization. Work already completed is kept and reported; the run ends with partial results clearly marked in the log and summary.

NOTE Closing the window while a run is in progress prompts you to stop it first; the program then closes once the run has safely halted.

18. Troubleshooting

Symptom	Likely cause	What to do
"The source folder does not exist"	Path typo, or a network share not currently reachable	Check the path; for a share that needs a login, set Credentials (Section 13).
"The source and destination are the same folder"	Both paths point to the same location	Set two different folders; DDiff never syncs a folder onto itself.
Many copies fail and the run aborts	Wrong destination, no permission, or the disk is full	Check destination path, write permission, and free space, then re-run.
A file shows as Failed	The file was locked, read-only in a way that could not be cleared, or the path was too long	Close any program using the file and re-run; check the log line for the exact reason.
Purge did not remove a folder	The folder still contained files that were excluded or newly created	Re-run; DDiff removes a directory only once it is empty.

19. Important Cautions

WARNING Synchronize with Purge on can delete up to 100% of the destination. Verify both paths before confirming.

CAUTION Deletes are not sent to the Recycle Bin. Use Delayed deletion when you want a safety net.

NOTE The read-only attribute is cleared before an overwrite or delete, so read-only files can be handled.

NOTE DDiff does not check for free space on the destination. Ensure there is enough room before synchronizing.

DDiff GUI is provided “as is”, without any warranty, expressed or implied. Use at your own risk.