

MCZ Synch

Compare two copies of a folder and bring them in step — User Manual for Linux

This manual describes **MCZ Synch in MCZ version 1.9.18**.

Issued 16 November 2026.

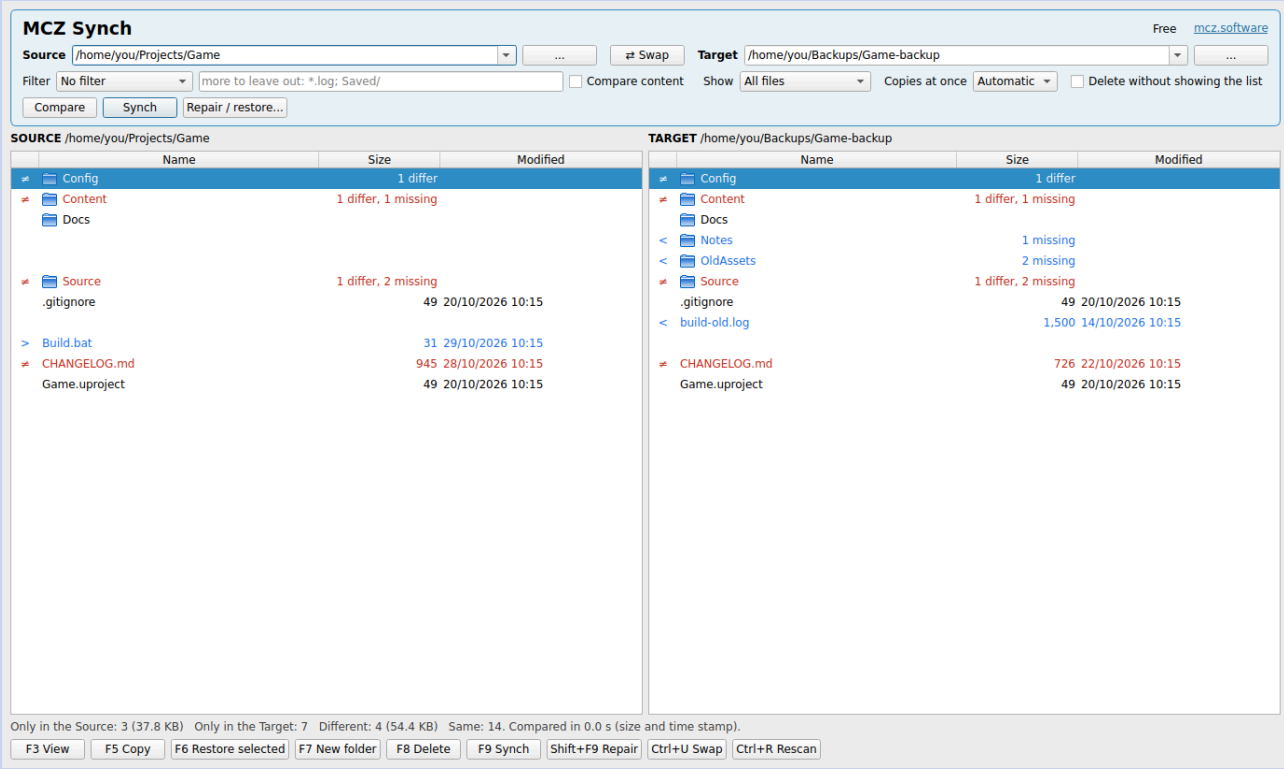
MCZ Synch is part of MCZ. To check which version you have installed, run `mcz version` in a terminal, or read the title bar of the MCZ window, which shows `MCZ: Version 1.9.18`. If the numbers differ from this page, fetch the manual that matches your software from the release it came with. Everything else about MCZ (archives, backups, profiles, the job queue) is in the *MCZ User Manual*; copying between PCs is in the *MCZ Copy User Manual*.

Contents

1. What MCZ Synch is
    1.1 Source and Target
    1.2 Free
2. Installing and starting
    2.1 Installing
    2.2 Where MCZ Synch keeps its settings
3. The window
    3.1 The top bar
    3.2 The two panels
    3.3 Colours and marks
    3.4 Moving around
4. Comparing
    4.1 How two files are compared
    4.2 Compare content
    4.3 Filters
    4.4 Links, unreadable files and other cases
5. Synch
    5.1 What a Synch does
    5.2 The delete list
    5.3 Delete without showing the list
6. Repair / restore
    6.1 Restoring a damaged project from a backup
    6.2 Restoring single files and folders
7. Single actions: F3 to F8
8. The synch screen
    8.1 Progress and log
    8.2 Cancel
    8.3 Copies at once
    8.4 How files are written
9. Safety
10. The command line
11. Troubleshooting

1. What MCZ Synch is

MCZ Synch compares two folders that hold copies of the same project, for example the copy you work in and a backup of it, and shows them side by side in two panels. Everything that differs is coloured: blue when a file or folder is on one side only, red when it is on both sides but different. One click then brings the second copy in step with the first (**Synch**), or puts back files from a good backup after something has damaged the project (**Repair / restore**).



A project (Source, left) compared with its backup (Target, right). Blue: on one side only. Red: on both sides but different.

MCZ Synch works on whole files: a file is always taken whole from one side, never merged line by line. It works on any folder this PC can open: local disks, USB drives and mounted network shares. It does not connect to other PCs by itself; for that, use MCZ Copy.

Comparing never changes anything. Files only change when you start a Synch, a Repair / restore or one of the single actions in section 7, and each of them shows what it is about to do first.

1.1 Source and Target

The left panel is the **Source** and the right panel is the **Target**. Files always go from the Source to the Target: a Synch or a Repair / restore changes the Target and never the Source. The panel headings and every question MCZ Synch asks name the two sides, so there is never doubt about which way files will go.

To copy the other way, press **Swap** (or Ctrl+U): the two folders change places and are compared again. There is no right-to-left direction anywhere else.

Which is which? For keeping a backup up to date, your working copy is the Source and the backup is the Target. For repairing a damaged project, it is the other way round: the good backup is the Source and the damaged project is the Target (section 6).

1.2 Free

MCZ Synch is free. It needs no premium key and has no trial period: every part of it works from the first day, with or without a key, and it never asks for one.

2. Installing and starting

2.1 Installing

MCZ Synch is in the MCZ package:

```
sudo apt install ./mcz_1.9.18_amd64.deb
```

This installs `/usr/bin/mcz-synch` with a desktop entry (*MCZ Synch*) and this manual under `/usr/share/doc/mcz/`. The tarball `mcz-1.9.18-Linux.tar.gz` holds the same program for other distributions; it needs Qt 6.

Start it from the desktop menu, or run `mcz-synch` in a terminal. It needs no root rights and no service.

2.2 Where MCZ Synch keeps its settings

MCZ Synch remembers the last folders used on each side (the 12 most recent, in the drop-down lists of the two path boxes), the filter, the extra names to leave out, the Show choice, Compare content, Copies at once and the window size.

What	Where
Settings	`~/.config/MCZ/MCZ Synch.conf`
Program, this manual	`/usr/bin/mcz-synch`, `/usr/share/doc/mcz/`

3. The window

Two panels, Source on the left and Target on the right, always showing the same folder inside each copy.

3.1 The top bar

Control	What it does
Source, **Target**	The two folders. Type or paste a path, pick a recent one from the drop-down, or press **...** to choose a folder.
Swap	Exchanges the Source and the Target and compares again (Ctrl+U).
Filter	Leaves out build output and tool folders: *No filter*, *Visual Studio / C++*, *Unreal Engine*, *Git*, *Web (node)*. The box beside it takes more names to leave out (section 4.3).
Compare content	Also reads files of the same size byte for byte. Off by default (section 4.2).
Show	*All files*, *Differences only*, *Only in the Source* or *Only in the Target*. *Differences only* hides files and folders that are the same, which is the useful view on a big project.
Copies at once	How many files a Synch or Repair / restore copies at the same time (section 8.3).
Delete without showing the list	A Synch then deletes what is only in the Target without showing the delete list first (section 5.3).
Compare	Compares the two folders (again). Ctrl+R does the same.
Synch	Brings the Target in step with the Source (section 5). F9 does the same.
Repair / restore...	Puts back files from the Source, older copies included, after a list (section 6). Shift+F9 does the same.

The line under the panels gives the totals of the last compare, for example *Only in the Source: 14 (3.2 MB) Only in the Target: 2 Different: 9 (1.1 MB) Same: 41 220*, and how long it took, or *The folders are the same*. The buttons below it do the same as the keys F3 to F9 (section 7). **mcz.software** at the top right opens the MCZ website.

3.2 The two panels

Each panel lists the folders and then the files in the current folder, with their **Name**, **Size** (in bytes) and **Modified** date. The heading above each panel gives the folder it shows.

- **The panels move together.** Both always show the same place inside each copy: open `src` on the left and the right panel shows the Target's `src`. A folder that is on one side only opens with the other panel empty.

- **Lines are level.** Both panels list every name found on either side, in the same order. Where a name is on one side only, the other panel shows an empty line, so a file is always level with its partner. The panels scroll together.
- **Folders** show in their Size column what is below them: *3 differ, 1 missing* counts the files anywhere inside that are different and those on one side only. A folder whose whole content is the same shows nothing there.
- Click a column heading to sort both panels by name, size or date; click it again to reverse the order. Folders always come first.

3.3 Colours and marks

Look	Mark	Meaning
Blue	`>` in the Source, `<` in the Target	Only on this side. The other panel shows an empty line.
Red, in both panels	`≠`	On both sides, but different (section 4.1). A **folder** is red when anything inside it is different or on one side only.
Orange (yellow in the dark theme)	`!`	Could not be read (locked, no permission). Never counted as the same, never deleted.
Normal text		The same on both sides. A normal folder is the same all the way down.
Italic, with ↪		A link (section 4.4).

Colour is never the only signal: the narrow mark column at the left of each panel says the same, so MCZ Synch can be used by people who do not tell red from blue, and in a black-and-white printout. Hold the mouse over a coloured line for the reason, for example *Only in the Target* or *different size*. The colours follow the light or dark theme of the desktop.

3.4 Moving around

Key or mouse	What it does
Double-click a folder, or Enter	Opens the folder in both panels.
Double-click `..`, or Backspace	Goes up one folder in both panels.
Tab	Moves the keyboard focus to the other panel. F7 and F8 act in the panel with the focus.
Click, Ctrl+click, Shift+click	Selects one or more lines, for F5, F6 and F8.
Ctrl+R	Compares again.
Ctrl+U	Swaps the Source and the Target.

4. Comparing

Choose the two folders and press Compare.

MCZ Synch reads both folders at the same time, with several threads each, and shows the result when it has finished. The line under the panels counts the files found while it reads. A project of 100 000 files on a fast local disk takes a few seconds; on a network share it takes longer, as every folder has to be asked for its list. No file is opened unless **Compare content** is ticked.

MCZ Synch compares again by itself before every Synch, Repair / restore and single action, so it always works from what is on disk at that moment, never from an old result.

4.1 How two files are compared

Files are matched by their path inside each folder. Names are matched with letter case, as Linux does. For each name that is on both sides:

1. **Different size**: red.
2. **Same size, modified dates more than 2 seconds apart**: red, as one copy has changed. The 2 seconds allow for USB sticks and some network drives, which store dates rounded to 2 seconds.
3. **Same size, same date (within 2 seconds)**: the same.

4.2 Compare content

A program can change a file without changing its size or its date. To catch that, tick **Compare content**: every pair of files of the same size is then read side by side and compared byte for byte. The first difference makes the file red, and a pair whose bytes match is the same even when the dates differ (a file copied without its date). The line under the panels shows *Comparing content: 1 200 of 8 000 files* while it works.

This reads every such file in full on both sides, so on a project of many gigabytes it is slow, which is why it is off by default. Turn it on for a final check, not for every compare.

4.3 Filters

Build output and tool folders change all the time and are not worth comparing. The **Filter** list leaves them out; a folder that is left out is not read at all, which also makes the compare faster.

Filter	Leaves out
Visual Studio / C++	`.vs`, `bin`, `obj`, `x64`, `x86`, `Debug`, `Release`, `ipch` folders; `*.user`, `*.vsidx`, `*.suo` files
Unreal Engine	`Binaries`, `Intermediate`, `Saved`, `DerivedDataCache`, `.vs` folders
Git	the `.git` folder
Web (node)	`node_modules`, `dist` folders

The box beside the list takes more names, separated by `;`. `*` and `?` work as wildcards, and a name ending in `/` means folders only: `*.log; *.tmp; Saved/`. A name matches at any depth. What is left out is neither compared nor copied nor deleted.

4.4 Links, unreadable files and other cases

- **Links** (symbolic links) are never followed, so nothing outside the two folders is compared and a link cannot lead round in a circle. A link on both sides is the same when both point to the same place. A Synch never copies or deletes a link.
- **Unreadable** files and folders (locked, or no permission) are orange (yellow in the dark theme). They are never treated as the same and never deleted. When a Source folder cannot be read, nothing below the matching Target folder is deleted either.
- **A file on one side and a folder of the same name on the other** is red, with a note. MCZ Synch leaves it alone: sort it out by hand with F8.
- Temporary files left by a cancelled or interrupted synch (ending in `.mczsynch.tmp`) are not listed, and are removed from the Target by the next Synch.

5. Synch

Make the Target like the Source: F9, or the Synch button.

5.1 What a Synch does

In the Source	In the Target	What Synch does
a file or folder (blue)	missing	**Copied** to the Target.
a file (red)	an older copy	The Source copy **replaces** the Target copy.
a file (red)	the same date, another size	The Source copy **replaces** the Target copy.
a file (red)	a newer copy	**Kept**: a Synch never puts an older copy over a newer one. The result says how many were kept; use Repair / restore (section 6) to put the Source copy back anyway.
missing	a file or folder (blue)	**Deleted**, after the delete list (5.2).

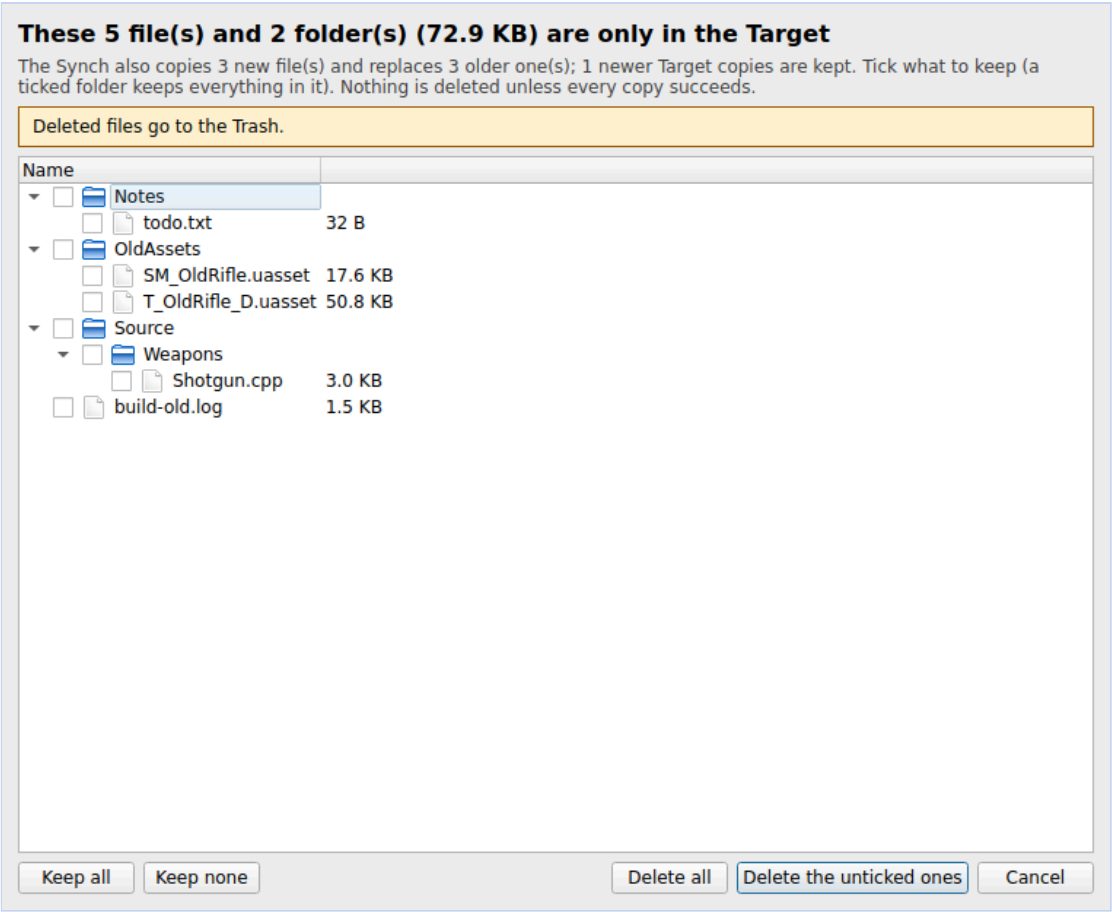
With **Compare content** ticked, a file whose date and size match but whose bytes differ is replaced by the Source copy too.

Replaced and deleted files go to the Trash, so they can be got back. On a network drive there is no Trash, so there they are deleted for good; the delete list says so.

Copies come first. Deletes happen only after every copy has succeeded, so a copy that fails never leaves the Target with less than it had. When the Target already matches, MCZ Synch says *The Target is already in step with the Source* and does nothing.

5.2 The delete list

When the Target holds files or folders that are not in the Source, a Synch lists them before anything happens, grouped by folder as a tree, with their sizes. The heading counts the files and folders that would be deleted (a folder that only holds them is shown in the tree too, but not counted), and the line below it what else the Synch will do (new files copied, older ones replaced, newer ones kept).



The delete list: tick what to keep. A ticked folder keeps everything in it.

Everything in the list is deleted unless you **tick it to keep it**. Ticking a folder keeps everything inside it. Space ticks or unticks the selected line.

Button	What it does
Keep all, **Keep none**	Ticks or unticks every line.
Delete the unticked ones	Starts the Synch; ticked files and folders stay in the Target.
Delete all	Asks once more, then starts the Synch and deletes everything in the list, ticked or not.
Cancel	Nothing is changed.

Keeping a file in the Target is fine: it stays where it is and the next Synch lists it again.

5.3 Delete without showing the list

For a regular Synch where you know the Target holds nothing of its own, tick **Delete without showing the list** in the top bar. A Synch then deletes everything that is only in the Target without asking. Deleted files still go to the Trash (for good on a network drive), and the safety rules of section 9 still apply.

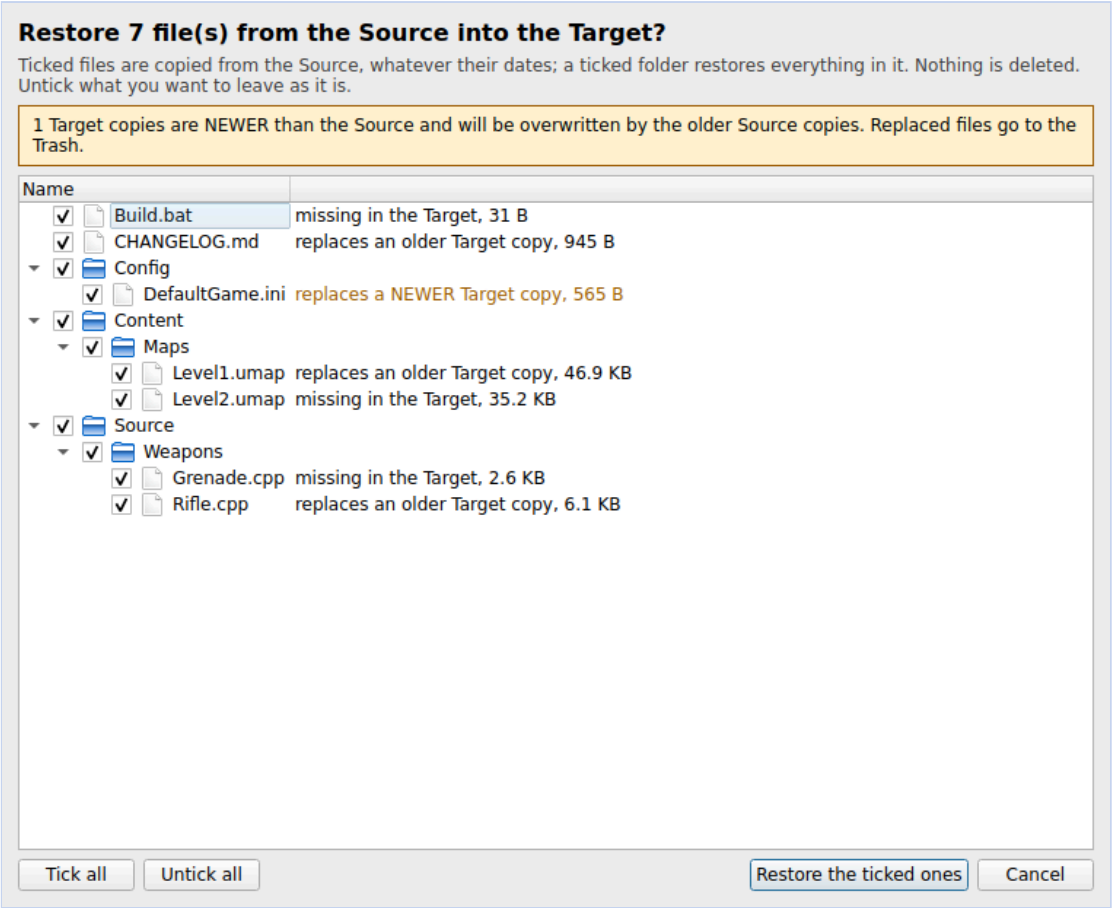
6. Repair / restore

Put back files from a good backup: Shift+F9, or Repair / restore....

6.1 Restoring a damaged project from a backup

A script that went wrong, a bad search and replace or a tool that rewrote files can damage a project. Every damaged file is then *newer* than its good copy in the backup, so a Synch from the backup would keep the damaged files. **Repair / restore** is for this case: older Source copies may overwrite newer Target copies.

1. Put the **good backup** as the Source (left) and the **damaged project** as the Target (right). Press Swap if they are the other way round.
2. Press **Repair / restore...** (Shift+F9). MCZ Synch compares again and lists every file that differs and every file the Target is missing, grouped by folder.
3. Untick what you want to leave as it is, and press **Restore the ticked ones**.



The Repair / restore list. Lines that will overwrite a newer Target copy are marked.

Every line starts ticked. Each one says what will happen: *missing in the Target*, *replaces an older Target copy*, or *replaces a NEWER Target copy*, in the warning colour (orange, or yellow in the dark theme), as that last kind overwrites work that may be newer than the backup. The warning at the top counts them. Ticking a folder restores everything in it that differs; **Tick all** and **Untick all** do what they say.

Repair / restore **never deletes**. Files that are only in the Target stay, as they may be junk left by the script or real work done since the backup; deleting them is a job for a Synch (section 5). Replaced files go to the Trash, or are gone for good on a network drive.

6.2 Restoring single files and folders

Select the files or folders to put back in either panel and press **F6** (*Restore selected*). The same list opens with only the selection, and a selected folder brings everything in it that differs. Everything else is left as it is.

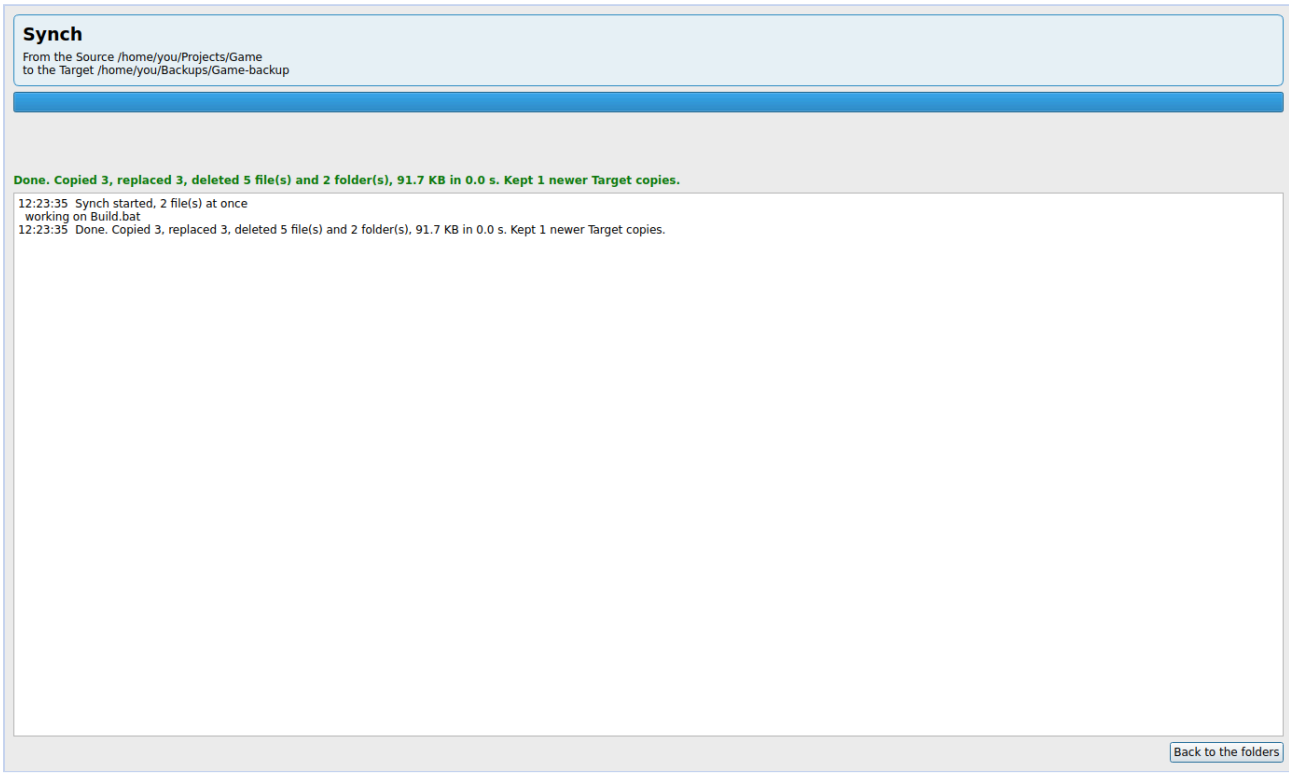
7. Single actions: F3 to F8

The keys work as in a two-panel file manager. Each one acts on the selected lines, and each one that changes files asks first.

Key	What it does
F3 View	Opens the selected file with its usual program, from the panel with the focus. Only to look: MCZ Synch never merges the inside of files.
F5 Copy	Copies the selection from the Source to the Target with the Synch rule: missing files are copied and older ones replaced, a newer Target copy is kept. It never deletes.
F6 Restore selected	Repair / restore of the selection only (section 6.2): older Source copies may overwrite newer ones.
F7 New folder	Makes a folder in the current folder of the panel with the focus.
F8 Delete	Deletes the selection in the panel with the focus, to the Trash (for good on a network drive). Delete does the same. In the Source panel it warns that this changes the Source, which a Synch never does.
F9, **Shift+F9**	Synch (section 5), Repair / restore (section 6).

8. The synch screen

When a Synch, a Repair / restore or an F5 or F6 copy starts, the panels make way for the synch screen. When it ends, press **Back to the folders**: the two folders are compared again, so the panels show the result.



The synch screen at the end of a Synch.

8.1 Progress and log

The screen shows the two folders, a progress bar with the files and the amount copied so far of the total, the speed in MB/s and files per second, the time left, and the file being worked on. The log below says when the job started and how many files it copies at once, and at the end gives the result: what was copied, replaced and deleted, how many newer Target copies were kept and how many Target-only items were left in place, the size and the time taken. Files that failed are listed with the reason, for example a file that is open in another program. One failure never stops the rest.

8.2 Cancel

Cancel stops the job after the files being copied at that moment. Their temporary files are removed, so nothing half-written is left behind. **A cancel does not undo**: files already copied, replaced or deleted stay that way, and the log says what was done. If a cancel comes before the deletes of a Synch, nothing is deleted. Closing the window during a job asks first, and then cancels the same way.

8.3 Copies at once

Copying many small files is mostly waiting: every file needs to be opened, created, closed, given its date and (on Windows) checked by the virus scanner, and on a network drive each of those is a round trip. Copying several files at the same time overlaps that waiting and makes a project of many small files several times faster. **Copies at once** sets how many:

- **Automatic** (the default): 8, or 2 when the Source or the Target is on a spinning hard disk, which would otherwise spend its time seeking.
- 1, 2, 4, 8, 16 or 32: a fixed number. Try fewer for a slow USB stick or a busy network drive, more for a fast local disk full of small files.

8.4 How files are written

Each file is first written under a temporary name ending in `.mczsynch.tmp` in its Target folder, then given the Source file's modified date and renamed into place. A cancel, a crash or a power cut therefore never leaves half a file under the real name. Temporary files that are left behind by a crash are removed by the next Synch.

Copied files keep the Source's modified date, so the next compare finds them the same. Folders that are missing in the Target are made first.

9. Safety

A synch to the wrong place can do a lot of damage in a short time, so MCZ Synch refuses some folders and some jobs outright:

- **Drive roots and system folders**, as the Source or the Target: `/`, the root of a mounted disk, system folders such as `/etc`, `/usr`, `/bin` and `/var`, `/home` and a user's own home folder. Choose a folder inside them instead, such as the project folder.
- **The same folder twice**, or one folder inside the other.
- **An empty Source** when the Synch would delete files in the Target: one wrong path must never empty a project. A Source that cannot be read stops the compare.
- **A Target that does not exist**: make it first (MCZ Synch does not guess at a typing error).

Within the two folders:

- Links are never followed, copied or deleted (section 4.4).
- Files that hold sign-in details, such as SSH keys and credential stores, are never deleted.
- Nothing below a Source folder that could not be read is deleted from the Target.
- Deleted and replaced files go to the Trash, except on network drives, where every list and question says that they are deleted for good.

10. The command line

The same compare, Synch and Repair / restore are available through `mcz`, for scripts, scheduled tasks and PCs without a desktop. They use the same rules and the same refusals as the window.

Command	What it does
`mcz synch-diff <source> <target> [--content] [--filter vs|unreal|git|web] [--exclude <pattern>] [--all]`	Lists what differs: `>` only in the Source, `<` only in the Target, `≠` different, `!` unreadable, then the totals. `--all` lists the files that are the same too.
`mcz synch <source> <target> [--keep <path>]... [--delete-all | --no-delete] [-n <1..32>] [--dry-run] [--yes]`	Synch (section 5). Prints the plan, then asks whether to delete what is only in the Target (no = copy only).
`mcz restore <source> <target> [<path>...] [-n <1..32>] [--dry-run] [--yes]`	Repair / restore (section 6) of everything that differs, or of the paths given. Never deletes.

Option	Meaning
`--content`	Compare content (section 4.2).
`--filter`, `--exclude`	A filter of section 4.3, and more names to leave out; both can be given more than once.
`--keep <path>`	A Target-only file or folder (path inside the Target) that a Synch must not delete; can be given more than once.
`--delete-all`	Deletes what is only in the Target without asking (like 5.3).
`--no-delete`	Copies only; deletes nothing.
`-n`, `--workers`	Copies at once, 1 to 32 (section 8.3). Without it: automatic.
`--dry-run`	Prints the plan and changes nothing.
`--yes`, `-y`	Does not ask. With `synch` this deletes the Target-only items too, unless `--no-delete` is given.

```
mcz synch-diff ~/Projects/Game /media/backup/Game --filter unreal
mcz synch ~/Projects/Game /media/backup/Game --filter unreal --dry-run
mcz synch ~/Projects/Game /media/backup/Game --filter unreal --keep Notes/todo.txt
mcz restore /media/backup/Game ~/Projects/Game Source/Weapons Config/DefaultGame.ini
```

Paths after `restore` and `--keep` are relative to the two folders and may use `/` or `\`. Exit codes: `synch-diff` returns **0** when the folders are the same, **1** when they differ and **2** on an error, so a script can test it; `synch` and `restore` return **0** on success and **1** on a refusal, a failure or a cancel, with the reason printed, and **2** for a mistake in the command. Ctrl+C cancels as in section 8.2.

11. Troubleshooting

Symptom	Cause and remedy
the Source cannot be synched or *the Target cannot be synched*, with a reason	The folder is a drive root, a system folder or a profile folder (section 9). Choose the project folder inside it.
the Source and the Target are the same folder, or one is inside the other	Check the two paths; a mapped drive or a link can make two different-looking paths the same folder.
the Source ... is empty: a Synch would delete everything in the Target, so it is refused	The Source path is probably wrong, or a drive is not connected. Check it. To empty a folder on purpose, delete its content with F8.
The Target is already in step; 3 newer Target copies are kept	Those files were changed in the Target after the Source copy. To put the Source copies back anyway, use Repair / restore or F6.
A file stays red after a Synch	Its Target copy is newer (kept, as above), it is a file on one side and a folder on the other (4.4), or its copy failed: the log on the synch screen gives the reason.
Nothing was deleted, because not every copy succeeded	Some copies failed (listed in the log), so the Synch did not delete anything. Close the program holding the file open, or fix the permission, and synch again.
Files that are the same are shown red	Their dates differ by more than 2 seconds, for example after copying them with a tool that does not keep dates. Tick **Compare content** to compare their bytes instead; a Synch then leaves matching files alone.
Orange or yellow lines (*could not be read*)	A file is locked by another program or you have no permission to read it. Close the program, or check the folder's permissions, and press Compare again.
The compare takes long	Choose a filter (4.3) so build output is not read, and leave **Compare content** off unless you need it.
Copying to a USB stick or network drive is slow	Try fewer **Copies at once** (8.3).
A deleted file is not in the Trash	The Target is on a network drive, where deletes are for good (section 9).

MCZ Synch in MCZ 1.9.18 for Linux, 16 November 2026. Source and releases: github.com/peterbobwalton/Mcz. MCZ Synch is free; see `LICENSE.txt`.