# rsync QuickRef

rsync has no built-in date filter. Select files with `find` and hand the list to `--files-from`.

## Move files older than a date

```
# 1. build the list -- paths must be relative to the source root
cd /path/src
find . -type f ! -newermt '2026-01-01 00:00:00' -print0 > /tmp/oldfiles

# 2. transfer
rsync -a --from0 --files-from=/tmp/oldfiles . /path/dst
```

`! -newermt` is strictly-before (negated "modified at or after"). Do not use `-mtime`, which is day-granular and shifts with the current clock.

## Gotchas

- Paths in the list must be relative to the source dir. `find /path/src` yields `/path/src/a.txt`, and rsync then looks for `/path/src/path/src/a.txt`. Run `find .` from inside the source, or use

  ```
  find . -printf '%P\0'
  ```

  .

- `-R` (relative) is implied by `--files-from`, so the directory structure is preserved automatically.
- `-a` does **not** imply `-r` when `--files-from` is used. Only matters when the list contains directories.
- `-print0` together with `--from0` survives spaces and newlines in filenames.

## Creation date vs mtime

POSIX has no portable creation time, and GNU `find` has no btime predicate, so `-newermt` filters on **mtime**.

To filter on birth time (ext4/xfs, via statx):

```
find . -type f -printf '%W\0%P\0' | while IFS= read -r -d '' b && IFS= read
-r -d '' f; do
  [ "$b" -gt 0 ] && [ "$b" -lt "$(date -d '2026-01-01' +%s)" ] && printf
'%s\0' "$f"
done > /tmp/oldfiles
```

`%W` prints 0 when the filesystem has no birth time, hence the guard. Check support first:

```
stat -c '%n btime=%W mtime=%Y' file
```

# Removing the source files afterwards

The safe route is to let rsync track it. Re-run the same command with `--remove-source-files`:

```
cd /path/src
rsync -a --remove-source-files --checksum --from0 --files-from=/tmp/oldfiles
. /path/dst
```

Each source file is removed only once that file is confirmed complete on the destination. Already-transferred files are not re-copied, but they are still deleted, so the command is idempotent and an interrupted run resumes cleanly.

Do **not** reach for `-n --remove-source-files` to preview the removals: it prints nothing about them. Verify separately.

## Verify, then delete from the list

Only correct once the destination is proven complete:

```
cd /path/src
# empty output == every listed file is already correct on dst
rsync -ain --checksum --from0 --files-from=/tmp/oldfiles . /path/dst

# then, only if the above printed nothing:
xargs -0 rm -f -- < /tmp/oldfiles
```

Non-empty output lists the files that still differ; do not delete yet. This route is all-or-nothing and trusts a list that may have gone stale, so prefer `--remove-source-files`.

`--checksum` matters here because default rsync compares only size and mtime, so same-size corruption at an identical mtime reads as “up to date”.

# Prune empty directories

rsync never removes directories, so clean them up afterwards:

```
find . -mindepth 1 -type d -empty -delete
```

# True moves

On the same filesystem, `find -exec mv` avoids the copy entirely. Across filesystems, or when a preview is wanted first, rsync is the better tool.

From:
https://memex.kyaruc.moe/ - **kyaruc memex**

Permanent link:
**https://memex.kyaruc.moe/quickref:rsync?rev=1795060672**

Last update: **2026-11-19 Thu 03:57**