

Files commands



- [File operations](#)
 - [files:cleanup](#)
 - [files:copy](#)
 - [files:delete](#)
 - [files:get](#)
 - [files:mount:list](#)
 - [files:mount:refresh](#)
 - [files:move](#)
 - [files:put](#)
 - [files:mkdir](#)
 - [files:touch](#)
 - [files:reminders](#)
 - [files:repair-tree](#)
 - [files:sanitize-filenames](#)
 - [files:scan](#)
 - [files:scan-app-data](#)
 - [files:transfer-ownership](#)
 - [files:windows-compatible-filenames](#)
- [Object store](#)
 - [files:object:list](#)
 - [files:object:get](#)
 - [files:object:info](#)
 - [files:object:put](#)
 - [files:object:delete](#)
 - [files:object:orphans](#)
- [Preview](#)
 - [preview:cleanup](#)
- [Trash bin](#)

- [trashbin:size](#)
- [trashbin:cleanup](#)
- [trashbin:expire](#)
- [File versions](#)
 - [versions:cleanup](#)
 - [versions:expire](#)
- [File sharing](#)
 - [share:list](#)
 - [sharing:cleanup-remote-storages](#)
 - [sharing:delete-orphan-shares](#)
 - [sharing:expiration-notification](#)
 - [sharing:fix-share-owners](#)
- [Federation sync](#)
 - [federation:sync-addressbooks](#)
- [Files external](#)
 - [files_external:list](#)
 - [files_external:create](#)
 - [files_external:config](#)
 - [files_external:applicable](#)
 - [files_external:option](#)
 - [files_external:verify](#)
 - [files_external:delete](#)
 - [files_external:scan](#)
 - [files_external:export](#)
 - [files_external:import](#)
 - [files_external:backends](#)
 - [files_external:notify](#)
 - [files_external:dependencies](#)
- [Trashbin](#)
 - [trashbin:cleanup](#)
 - [trashbin:expire](#)
 - [trashbin:restore](#)
 - [trashbin:size](#)
- [Versions](#)

- [versions:cleanup](#)
- [versions:expire](#)
- [Integrity check](#)
 - [integrity:check-app](#)
 - [integrity:check-core](#)
 - [integrity:sign-app](#)

The `occ` commands in this section cover day-to-day file management, storage maintenance, and sharing administration. They let you scan and repair the file cache, move or delete files on behalf of users, manage trash and version history, configure external storage mounts, inspect object store contents, and verify app integrity from the command line.

Several sections require specific apps to be enabled. All of the following are shipped with Nextcloud and can be enabled in the Apps menu:

- **Trash bin** commands require the **Deleted files** app (`files_trashbin`) — enabled by default
- **File versions** commands require the **Versions** app (`files_versions`) — enabled by default
- **Federation sync** requires the **Federation** app (`federation`) — enabled by default
- **External storage** commands require the **External storage support** app (`files_external`) — not enabled by default

File operations

files	
files:cleanup	remove file cache entries with no matching storage entry
files:copy	copy a file or folder
files:delete	delete a file or folder
files:get	get the contents of a file
files:mount:list	list all mounts for a user
files:mount:refresh	refresh the list of mounts for a user
files:move	move a file or folder
files:put	write content to a file
files:mkdir	create a new folder
files:touch	update the modified date of a file or folder
files:reminders	list file reminders
files:repair-tree	repair malformed filesystem tree structures

files:sanitize-filenames	rename files that do not match the current filename
constraints	
files:scan	rescan the filesystem
files:scan-app-data	rescan the AppData folder
files:transfer-ownership	transfer all files and shares from one user to another
files:windows-compatible-filenames	toggle enforcement of Windows-compatible filenames

files:cleanup

Remove all file cache entries that have no matching entry in the storage table. Run this after removing orphaned storage records or after a migration that left the file cache in an inconsistent state:

```
sudo -E -u www-data php occ files:cleanup
```

files:copy

Copy a file or folder within Nextcloud. Source and target accept either a Nextcloud path or a numeric file ID:

```
sudo -E -u www-data php occ files:copy /layla/files/Photos \
/layla/files/Photos-backup
```

If the target path already exists and is a folder, the source is copied into it (standard cp behaviour). Use `--no-target-directory` / `-T` to overwrite the target folder instead:

```
sudo -E -u www-data php occ files:copy -T /layla/files/Photos \
/layla/files/Photos-backup
```

If the target exists the command asks for confirmation. Use `--force` / `-f` to skip the prompt and suppress type-mismatch warnings.

files:delete

Delete a file or folder:

```
sudo -E -u www-data php occ files:delete /layla/files/Documents/old-draft.pdf
```

The command asks for confirmation before deleting. Use `--force` / `-f` to skip the prompt.

If the path points to the root of a share received by a user, the command asks whether to unshare (remove the user's access) rather than delete the underlying file:

```
sudo -E -u www-data php occ files:delete /layla/files/Shared-Folder  
/layla/files/Shared-Folder in a shared file, do you want to unshare  
the file from layla instead of deleting the source file? [Y/n]
```

When the file is accessible by multiple users, the command lists all affected users and asks for confirmation before proceeding.

If the trashbin is enabled, you can use `--skip-trash` to bypass the trashbin and permanently delete the files directly.

files:get

Download a file from Nextcloud to a local path. The source argument accepts either a Nextcloud path or a numeric file ID:

```
sudo -E -u www-data php occ files:get /layla/files/Documents/report.pdf \  
/tmp/report.pdf
```

Omit the output path (or pass `-`) to write to standard output:

```
sudo -E -u www-data php occ files:get /layla/files/Documents/report.pdf -
```



Writing binary files to a terminal can corrupt your terminal session. Nextcloud will refuse to write binary content to STDOUT unless you explicitly pass `-` as the output path.

files:mount:list

List all registered mounts for a user, including mount point, storage ID, provider, and whether the mount is still actively provided:

```
sudo -E -u www-data php occ files:mount:list layla  
/layla/: local::home/nextcloud/data/layla/files  
- provider: OC\Files\Mount\LocalRootMountProvider
```

- storage id: 1
- root id: 1

Stale mounts (registered in the cache but no longer provided) are marked with `registered but no longer provided`.

files:mount:refresh

Re-register all currently active mounts for a user in the mount cache. Useful after mount configuration changes that were not automatically picked up:

```
sudo -E -u www-data php occ files:mount:refresh layla  
Registered 3 mounts
```

files:move

Move a file or folder within Nextcloud:

```
sudo -E -u www-data php occ files:move \  
/layla/files/Documents/draft.pdf \  
/layla/files/Documents/Archive/draft.pdf
```

If the target already exists, the command asks for confirmation. Use `--force` / `-f` to skip the prompt. If source and target are different types (file vs. folder), the target is deleted before the move; the command refuses if the target is a mount root or is not deletable.

files:put

Upload a local file to Nextcloud. The target argument accepts either a Nextcloud path or a numeric file ID of an existing file:

```
sudo -E -u www-data php occ files:put /tmp/report.pdf \  
/layla/files/Documents/report.pdf
```

Read from standard input by passing `-` as the source:

```
cat /tmp/report.pdf | sudo -E -u www-data php occ files:put - \  
/layla/files/Documents/report.pdf
```

Any missing parent folder for the specified destination will be created automatically.

files:mkdir

Create a new empty folder, parent folders will automatically be created if needed:

```
sudo -E -u www-data php occ files:mkdir /layla/files/Documents/Reports
```

files:touch

Update the last modified date of a file or folder. The target argument accepts either a Nextcloud path or a numeric file ID of an existing file:

```
sudo -E -u www-data php occ files:touch /layla/files/Documents/report.pdf
```

Instead of using the current time, you can specify the modified date to use with the `--date` option. Accepted formats are: ISO8601, "YYYY-MM-DD" and Unix time in seconds.:

```
sudo -E -u www-data php occ files:touch /layla/files/Documents/report.pdf \
--date 2026-07-15
```

files:reminders

List all file reminders across all users, or for a specific user:

```
sudo -E -u www-data php occ files:reminders
```

User	File Id	Path	Due Date (UTC)
layla	42	/layla/files/Documents/report.pdf	2026-05-01T09:00:00+00:00

```
sudo -E -u www-data php occ files:reminders layla
```

Use `--output=json_pretty` for machine-readable output.

files:repair-tree

Repair entries in the file cache whose stored path does not match the path derived from their parent. This can cause files to appear in the wrong location: listing a folder shows the file, but

accessing it by path fails. The command resets each affected entry to its correct path:

```
sudo -E -u www-data php occ files:repair-tree
```

files:sanitize-filenames

Rename files that do not comply with the current filename constraints (for example, after enabling [Windows-compatible filenames](#)). Run for all users or a specific user:

```
sudo -E -u www-data php occ files:sanitize-filenames
sudo -E -u www-data php occ files:sanitize-filenames layla
```

Invalid characters are replaced with a space, underscore, or dash — whichever is allowed by the current constraints. Use `--char-replacement` to specify the replacement character when none of those defaults are permitted:

```
sudo -E -u www-data php occ files:sanitize-filenames \
--char-replacement=_ layla
```

Use `--dry-run` to preview the renames without making changes:

```
sudo -E -u www-data php occ files:sanitize-filenames --dry-run
```

files:scan

Scan for new or changed files and update the file cache. Run for all users, a specific user, or a specific path:

```
sudo -E -u www-data php occ files:scan --all
sudo -E -u www-data php occ files:scan layla
sudo -E -u www-data php occ files:scan layla fred
```

Use `--path` to limit the scan to a specific directory. The path must include the user ID and `files/`:

```
sudo -E -u www-data php occ files:scan --path="/layla/files/Photos"
```

The `--path`, `--all`, and `[user_id]` options are mutually exclusive; only one may be given at a time.

Additional options:

- `--generate-metadata` — generate metadata (e.g. EXIF data) for scanned files
- `--unscanned` — only scan files marked as not yet fully scanned (useful when triggering the scan that the background job would otherwise run)
- `--shallow` — do not scan folders recursively
- `--home-only` — skip external storages and shares, scan only the home storage
- `--quiet` — suppress output; without this option, statistics are shown after the scan
- `--output=json_pretty` — machine-readable output

Use `-v` to see each file as it is processed. Verbosity levels `-vv` and `-vvv` are silently capped to `-v`.

A background job runs every 10 minutes to scan files with an unknown size in the file cache, so routine rescans are handled automatically. Use this command when you need an immediate rescan — for example after copying files directly into the data directory, after a migration, or to investigate file cache inconsistencies. The background scan can be disabled by setting `'files_no_background_scan' => true` in `config.php`.

files:scan-app-data

Rescan the `appdata` directory and update the file cache for files shared between users (avatar images, file previews, cached CSS, etc.):

```
sudo -E -u www-data php occ files:scan-app-data
```

Limit the scan to a specific appdata subdirectory:

```
sudo -E -u www-data php occ files:scan-app-data preview
```

Unlike `files:scan`, there is no background job that automatically rescans appdata. Run this command manually when the appdata file cache may have become inconsistent; for example, after restoring appdata from a backup, after manually moving or copying files into the appdata directory, or when previews or avatars are not displaying correctly despite the underlying files being present on disk.

files:transfer-ownership

Transfer all files and shares from one user to another. Useful before removing a user account:

```
sudo -E -u www-data php occ files:transfer-ownership layla fred
```

The transferred files appear in a subdirectory inside the destination user's home folder.



Unless server-side encryption is enabled, the command initialises the destination user's file system if they have not yet logged in. If the destination user's data directory cannot be written to, the command reports: `unable to rename, destination directory is not writable`.

If the destination user's home is empty, use `--move` to move files directly into the root without creating a subdirectory:

```
sudo -E -u www-data php occ files:transfer-ownership --move layla fred
```

Transfer only a specific folder with `--path`:

```
sudo -E -u www-data php occ files:transfer-ownership \  
--path="Documents/Project" layla fred
```

Incoming shares (files shared with the source user) are not transferred by default because ownership remains with the original sharer. Transfer them with `--transfer-incoming-shares=1`:

```
sudo -E -u www-data php occ files:transfer-ownership \  
--transfer-incoming-shares=1 layla fred
```

Set `'transferIncomingShares' => true` in `config.php` to always transfer incoming shares. The command-line option takes precedence:

```
sudo -E -u www-data php occ files:transfer-ownership \  
--transfer-incoming-shares=0 layla fred
```

Users may also transfer files selectively via the web interface. See [user documentation](#) for details.

files:windows-compatible-filenames

Toggle enforcement of [Windows-compatible filenames](#):

```
sudo -E -u www-data php occ files:windows-compatible-filenames --enable  
sudo -E -u www-data php occ files:windows-compatible-filenames --disable
```

After enabling, run `files:sanitize-filenames` to rename any existing files that do not comply.

Object store

files	
files:object:delete	delete an object from the object store
files:object:get	get the contents of an object
files:object:info	get the metadata of an object
files:object:list	list all objects in the object store
files:object:orphans	list objects in the object store with no matching database entry
files:object:put	write a file to the object store

These commands operate directly on the underlying object storage (S3, Swift, etc.) used as Nextcloud's primary storage. They bypass the file cache and normal access controls.



These commands manipulate objects directly in the object store. Writing or deleting objects that belong to existing files will corrupt those files. Use them only for debugging or recovery, never for routine file management.

files:object:list

List all objects in the object store. Requires the configured object store to support metadata listing:

```
sudo -E -u www-data php occ files:object:list
```

+-----+-----+-----+-----+			
URN	Size	Last modified	
+-----+-----+-----+-----+			
urn:oid:1	512	2026-01-01T00:00:00+00:00	
urn:oid:42	204800	2026-04-01T12:00:00+00:00	
+-----+-----+-----+-----+			

Use `--bucket` / `-b` if the bucket cannot be determined from the config. Use `--output=json_pretty` for machine-readable output.

files:object:get

Download an object from the object store to a local file. Pass `-` as the output path to write to standard output:

```
sudo -E -u www-data php occ files:object:get urn:oid:42 /tmp/recovered.pdf
sudo -E -u www-data php occ files:object:get urn:oid:42 -
```

files:object:info

Show metadata (size, MIME type, last-modified date) for an object:

```
sudo -E -u www-data php occ files:object:info urn:oid:42
- size: 200 KB
- mimetype: application/pdf
- mtime: 2026-04-01T12:00:00+00:00
```

Use `--output=json_pretty` for machine-readable output.

files:object:put

Upload a local file to the object store under the given object name. Read from standard input by passing `-`:

```
sudo -E -u www-data php occ files:object:put /tmp/data.bin urn:oid:42
sudo -E -u www-data php occ files:object:put - urn:oid:42
```

If the object already corresponds to a file in the database, the command warns and asks for confirmation before overwriting. To update a file safely, use `files:put` with the file ID instead.

files:object:delete

Delete an object from the object store:

```
sudo -E -u www-data php occ files:object:delete urn:oid:42
```

If the object belongs to a file in the database, the command warns that deleting it will corrupt the file and shows the file ID so you can use `files:delete` instead for a clean removal.

files:object:orphans

List objects in the object store that have no matching entry in the file cache database. These are objects that Nextcloud no longer tracks and may be safe to remove after investigation:

```
sudo -E -u www-data php occ files:object:orphans
```

Use `--output=json_pretty` for machine-readable output. Requires the object store to support metadata listing.

Preview

```
preview
```

```
preview:cleanup
```

```
remove all generated preview files
```

preview:cleanup

Remove all generated preview files. Useful after changing preview configuration (sizes, quality, or supported file types), or on installations using object storage as primary storage where the preview folder cannot be deleted manually:

```
sudo -E -u www-data php occ preview:cleanup
```

```
Previews removed
```

After running this command, Nextcloud regenerates previews on demand as users access files.

See [Previews configuration](#) for preview settings.

Trash bin

```
trashbin
```

```
trashbin:cleanup
```

```
permanently delete all files in the trash for a user
```

```
trashbin:expire
```

```
expire trash items that exceed the configured retention
```

```
period
```

```
trashbin:size
```

```
show or set the target trash bin size
```

trashbin:size

Show the current trash bin size target:

```
sudo -E -u www-data php occ trashbin:size  
Default size: default (50% of available space)
```

Show the effective size for a specific user:

```
sudo -E -u www-data php occ trashbin:size --user layla  
default (50% of available space)
```

Set the global default target size. Accepts human-readable sizes:

```
sudo -E -u www-data php occ trashbin:size 10GB
```

“ ”

Changing the global default immediately triggers a cleanup of existing trash bins. A user's trash may temporarily exceed the configured size until the user next moves a file to trash.

Set a per-user target size:

```
sudo -E -u www-data php occ trashbin:size --user layla 2GB
```

trashbin:cleanup

Permanently delete all files in the trash bin for one or more users, or for all users:

```
sudo -E -u www-data php occ trashbin:cleanup layla  
Remove deleted files of layla  
  
sudo -E -u www-data php occ trashbin:cleanup layla fred  
sudo -E -u www-data php occ trashbin:cleanup --all-users
```

Use `--verbose` to see the amount of data freed per user. Either a user ID or `--all-users` must be given; the two cannot be combined.

trashbin:expire

Expire trash items that exceed the configured retention period, as defined by

`trashbin_retention_obligation` in `config.php`:

```
sudo -E -u www-data php occ trashbin:expire layla
sudo -E -u www-data php occ trashbin:expire
```

“ ”

This command only runs when a custom retention period is configured. If Nextcloud is set to auto-expiration (the default), the command exits with an informational message and does nothing; auto-expiration is handled by the background job instead.

See [Controlling file versions and aging](#) for retention configuration.

File versions

versions	
versions:cleanup	delete stored file versions
versions:expire	expire file versions that exceed the configured
retention period	

versions:cleanup

Delete stored file versions for one or more users, all users, or a specific path:

```
sudo -E -u www-data php occ versions:cleanup layla
Delete versions of layla

sudo -E -u www-data php occ versions:cleanup layla fred
sudo -E -u www-data php occ versions:cleanup
```

Use `--path` to limit deletion to a specific directory. The path must include the user ID and `files/`:

```
sudo -E -u www-data php occ versions:cleanup \
--path="/layla/files/Documents" layla
```

versions:expire

Expire file versions that exceed the configured retention period, as defined by

`versions_retention_obligation` in `config.php`:

```
sudo -E -u www-data php occ versions:expire layla
sudo -E -u www-data php occ versions:expire
```

“ ”

This command only runs when a custom retention period is configured. If Nextcloud is set to auto-expiration (the default), the command exits and does nothing; auto-expiration is handled by the background job instead.

See [Controlling file versions and aging](#) for retention configuration.

File sharing

sharing	
sharing:cleanup-remote-storages	clean up remote storage entries with no matching
federated share	
sharing:delete-orphan-shares	delete shares where the owner no longer has file access
sharing:expiration-notification	notify share initiators when a share expires the next
day	
sharing:fix-share-owners	fix share owner after a legacy transfer-ownership
operation	
share	
share:list	list available shares

share:list

List shares across the instance. Without options, lists all shares:

```
sudo -E -u www-data php occ share:list
```

id	file	source-path	type	owner	recipient	by
----	------	-------------	------	-------	-----------	----

```
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 42 | /layla/files/Documents/rep... | user | layla | fred | layla |
+-----+-----+-----+-----+-----+-----+-----+
```

Filter options:

- `--owner` — only shares owned by a specific user
- `--recipient` — only shares with a specific recipient
- `--by` — only shares initiated by a specific user
- `--file` — only shares of a specific file (path or file ID)
- `--parent` — only shares of files inside a specific folder
- `--recursive` — combine with `--parent` to include nested shares
- `--type` — filter by share type: `user`, `group`, `link`, `email`, `remote`, `room`, `deck`
- `--status` — only shares with a specific status

Use `--output=json_pretty` for machine-readable output.

sharing:cleanup-remote-storages

Remove `shared::` storage entries from the database that have no matching entry in the `shares_external` table. These orphaned entries are left behind when a federated share is removed without proper cleanup:

```
sudo -E -u www-data php occ sharing:cleanup-remote-storages
5 remote storage(s) need(s) to be checked
3 remote share(s) exist
deleting shared::abc123 [14] ... deleted 1 storage
```

Use `--dry-run` to preview what would be deleted without making changes:

```
sudo -E -u www-data php occ sharing:cleanup-remote-storages --dry-run
```

sharing:delete-orphan-shares

Delete shares where the owner has lost access to the shared file or the file no longer exists:

```
sudo -E -u www-data php occ sharing:delete-orphan-shares
/layla/files/Documents/report.pdf owned by layla
file still exists but the share owner lost access to it,
run occ info:file 42 for more information about the file
Delete 1 orphan shares? [y/N]
```

Use `--force` / `-f` to delete without prompting. Filter to shares by a specific user with `--owner` or to shares with a specific user with `--with`:

```
sudo -E -u www-data php occ sharing:delete-orphan-shares --owner layla
sudo -E -u www-data php occ sharing:delete-orphan-shares --with fred
```

sharing:expiration-notification

Send in-app notifications to share initiators whose shares expire the following day. Run this daily via a cron job to ensure timely notifications:

```
sudo -E -u www-data php occ sharing:expiration-notification
```

sharing:fix-share-owners

Fix the recorded owner of shares that were broken by a `transfer-ownership` operation performed on an older Nextcloud version. Use `--dry-run` to preview changes:

```
sudo -E -u www-data php occ sharing:fix-share-owners --dry-run
Share with id 7 (target: /fred/files/report.pdf) can be
updated to owner layla

sudo -E -u www-data php occ sharing:fix-share-owners
Share with id 7 (target: /fred/files/report.pdf) updated to owner layla
No broken shares detected
```

Federation sync

```
federation
federation:sync-addressbooks      synchronize address books of all federated clouds
```

federation:sync-addressbooks

Synchronize the shared address books of all federated Nextcloud servers. Federated servers share user address books so that usernames auto-complete in share dialogs. Run this command to trigger an immediate sync rather than waiting for the background job:

```
sudo -E -u www-data php occ federation:sync-addressbooks
```

Files external

files_external	
files_external:applicable	manage applicable users and groups for a mount
files_external:backends	show available authentication and storage backends
files_external:config	manage backend configuration for a mount
files_external:create	create a new mount configuration
files_external:delete	delete an external mount
files_external:dependencies	check for missing dependencies needed for mounting
external storages	
files_external:export	export mount configurations
files_external:import	import mount configurations
files_external:list	list configured admin or personal mounts
files_external:notify	listen for active update notifications for a configured
external mount	
files_external:option	manage mount options for a mount
files_external:scan	scan an external storage for changed files
files_external:verify	verify mount configuration

Manage Nextcloud's external storage mounts. Commands that read or write mount configuration operate on the same data as the **External Storages** admin settings page.

files_external:list

List configured mounts. Without arguments, lists admin-level mounts:

```
sudo -E -u www-data php occ files_external:list
```

ID	Mount Point	Storage	Auth. Type	Config	Status	Users
1	/shared/data	amazons3	builtin	valid	ok	all

List personal mounts for a specific user:

```
sudo -E -u www-data php occ files_external:list layla
```

Use `--output=json_pretty` for machine-readable output. Add `--show-password` to include credentials in the output.

files_external:create

Create a new mount configuration:

```
sudo -E -u www-data php occ files_external:create \  
  /shared/photos amazons3 builtin::builtin \  
  --config bucket=my-nextcloud-photos \  
  --config region=eu-central-1 \  
  --config key=AKIAIOSFODNN7EXAMPLE \  
  --config secret=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

Arguments:

- **mount-point** — the path inside Nextcloud (e.g. `/shared/photos`)
- **storage-backend** — the storage type identifier (e.g. `amazons3`, `sftp`, `smb`, `owncloud`)
- **auth-backend** — the authentication method (e.g. `builtin::builtin`, `password::sessioncredentials`)

Use `files_external:backends` to list all available storage and authentication backends.

For a personal mount, specify the user with `--user`:

```
sudo -E -u www-data php occ files_external:create \  
  /my-s3 amazons3 builtin::builtin \  
  --config bucket=layla-bucket \  
  --user layla
```

files_external:config

Read or write individual backend configuration options for a mount. Use the mount ID from `files_external:list`:

```
sudo -E -u www-data php occ files_external:config 1 get bucket  
my-nextcloud-photos  
  
sudo -E -u www-data php occ files_external:config 1 set bucket new-bucket-name
```

files_external:applicable

Manage which users and groups have access to a mount:

```
sudo -E -u www-data php occ files_external:applicable 1 --add-user layla
sudo -E -u www-data php occ files_external:applicable 1 --add-group milliways
sudo -E -u www-data php occ files_external:applicable 1 --remove-user layla
sudo -E -u www-data php occ files_external:applicable 1 --remove-group milliways
```

A mount with no users or groups assigned is available to all users.

files_external:option

Read or write mount options such as `enable_sharing` or `filesystem_check_changes`:

```
sudo -E -u www-data php occ files_external:option 1 get enable_sharing
sudo -E -u www-data php occ files_external:option 1 set enable_sharing true
```

files_external:verify

Test the configuration of a mount and report whether a connection can be established:

```
sudo -E -u www-data php occ files_external:verify 1

+-----+-----+
| Result | Message |
+-----+-----+
| success |         |
+-----+-----+
```

Pass additional configuration key-value pairs (e.g. credentials) with `--config` when the stored configuration is incomplete:

```
sudo -E -u www-data php occ files_external:verify 1 \
--config user=layla --config password=secret
```

files_external:delete

Delete a mount configuration:

```
sudo -E -u www-data php occ files_external:delete 1
```

The command asks for confirmation. Use `--yes` to skip the prompt.

files_external:scan

Scan an external storage for changed files and update the file cache. Useful for storage backends that do not support active change notifications:

```
sudo -E -u www-data php occ files_external:scan 1
```

For mounts that use session credentials, pass them with `--user` and `--password`:

```
sudo -E -u www-data php occ files_external:scan 1 \  
--user layla --password secret
```

files_external:export

Export all admin mounts to JSON:

```
sudo -E -u www-data php occ files_external:export > mounts.json
```

Export personal mounts for a specific user:

```
sudo -E -u www-data php occ files_external:export layla > layla-mounts.json
```

Use the exported JSON with `files_external:import` to replicate the configuration on another Nextcloud instance.

files_external:import

Import mount configurations from a JSON file produced by `files_external:export`:

```
sudo -E -u www-data php occ files_external:import mounts.json
```

Import as personal mounts for a specific user:

```
sudo -E -u www-data php occ files_external:import \  
--user layla layla-mounts.json
```

files_external:backends

List all available storage and authentication backends:

```
sudo -E -u www-data php occ files_external:backends
```

Filter by type:

```
sudo -E -u www-data php occ files_external:backends storage
sudo -E -u www-data php occ files_external:backends authentication
```

Use `--output=json_pretty` to inspect the full capability and configuration schema for each backend.

files_external:notify

Listen for active change notifications from an external mount that supports push-based update events (e.g. SMB with inotify):

```
sudo -E -u www-data php occ files_external:notify 1
```

This is a long-running process; run it under a process supervisor or in a dedicated systemd service.

files_external:dependencies

Check for any missing PHP extensions or system packages required to use the configured external storage backends:

```
sudo -E -u www-data php occ files_external:dependencies
```

Trashbin

“ ”

These commands are only available when the "Deleted files" app (`files_trashbin`) is enabled.

```
trashbin
trashbin:cleanup  remove deleted files
trashbin:expire   expire deleted files according to the configured retention policy
trashbin:restore  restore all deleted files according to the given filters
trashbin:size     configure or show the target trashbin size
```

trashbin:cleanup

Remove all deleted files for all users, or for specific users:

```
sudo -E -u www-data php occ trashbin:cleanup
sudo -E -u www-data php occ trashbin:cleanup layla fred
```

trashbin:expire

Apply the configured trashbin retention policy and remove files that have exceeded the maximum retention age. Runs for all users by default, or for specific users:

```
sudo -E -u www-data php occ trashbin:expire
sudo -E -u www-data php occ trashbin:expire layla
```

trashbin:restore

Restore deleted files according to the given filters. Restore all deleted files for all users:

```
sudo -E -u www-data php occ trashbin:restore --all-users
```

Restore deleted files for specific users:

```
sudo -E -u www-data php occ trashbin:restore layla
```

Use `--scope` to limit the restore to a specific scope; one of `user`, `groupfolders`, or `all` (default: `user`):

```
sudo -E -u www-data php occ trashbin:restore --scope groupfolders layla
```

Use `--since` and `--until` to limit the restore to files deleted within a given time window. Both options accept any format supported by PHP `strtotime`:

```
sudo -E -u www-data php occ trashbin:restore --scope all \  
--since "2026-08-01 11:55:22" --until "2026-08-02 01:33:00" layla
```

Use `--dry-run` to simulate the restore without making any changes:

```
sudo -E -u www-data php occ trashbin:restore --dry-run --all-users
```

“ ”

Use `-v` or `-vv` to see more detail about the restore process and why some files may be skipped.

trashbin:size

Show or configure the target trashbin size. When called without arguments, shows the current configured size:

```
sudo -E -u www-data php occ trashbin:size
```

Set the default trashbin size for all users:

```
sudo -E -u www-data php occ trashbin:size 10GB
```

Set the trashbin size for a specific user:

```
sudo -E -u www-data php occ trashbin:size --user layla 5GB
```

Versions

“ ”

These commands are only available when the "Versions" app (`files_versions`) is enabled.

versions

versions:cleanup delete file versions

versions:expire expire file versions according to the configured retention policy

versions:cleanup

Delete all file versions for all users, or for specific users. Use `--path` to limit deletion to a specific path:

```
sudo -E -u www-data php occ versions:cleanup
```

```
sudo -E -u www-data php occ versions:cleanup layla
```

```
sudo -E -u www-data php occ versions:cleanup layla --path="/files/Documents"
```

versions:expire

Apply the configured version retention policy and remove versions that have exceeded the maximum retention age. Runs for all users by default, or for specific users:

```
sudo -E -u www-data php occ versions:expire
```

```
sudo -E -u www-data php occ versions:expire layla
```

Integrity check

integrity

integrity:check-app check integrity of an app using a signature

integrity:check-core check core integrity using a signature

integrity:sign-app sign an app using a private key

Apps that carry a `Featured` tag must be code-signed by Nextcloud. Unsigned featured apps cannot be installed.

integrity:check-app

Check the integrity of an app against its signature:

```
sudo -E -u www-data php occ integrity:check-app contacts
```

Check all installed apps at once:

```
sudo -E -u www-data php occ integrity:check-app --all
```

When the app passes, the command produces no output (use `-v` to confirm). When it fails, each integrity error is listed with details. Use `--path` to point to a non-standard app location:

```
sudo -E -u www-data php occ integrity:check-app \  
--path=/var/www/nextcloud/apps/myapp myapp
```

Apps without a `signature.json` file are skipped with an informational message.

integrity:check-core

Check the integrity of Nextcloud core against its signature:

```
sudo -E -u www-data php occ integrity:check-core
```

integrity:sign-app

Sign an app with a private key before distribution. Requires the key and certificate obtained through the Nextcloud signing process:

```
sudo -E -u www-data php occ integrity:sign-app \  
--path=/path/to/app \  
--privateKey=/path/to/myapp.key \  
--certificate=/path/to/myapp.crt
```

See [Code Signing](#) in the Developer manual for the full signing process.

📄 #3

📄 cocobook-sync 📄 2026-10-28 16:27:24 UTC

📄 cocobook-sync 📄 2026-10-29 01:09:17 UTC