

# Controlling file versions and aging

---



- [During Version Creation](#)
  - [During the Regular Background Job](#)
  - [Background job](#)
- 

The Versions app (files\_versions) expires old file versions automatically to ensure that users don't exceed their storage quotas. This is the default pattern used to delete old versions:

- For the first second we keep one version
- For the first 10 seconds Nextcloud keeps one version every 2 seconds
- For the first minute Nextcloud keeps one version every 10 seconds
- For the first hour Nextcloud keeps one version every minute
- For the first 24 hours Nextcloud keeps one version every hour
- For the first 30 days Nextcloud keeps one version every day
- After the first 30 days Nextcloud keeps one version every week

The versions are adjusted along this pattern every time a new version is created. Nextcloud will always keep the latest version in each of the time windows.

The Versions app never uses more than 50% of the user's currently available free space. If the stored versions exceed this limit, Nextcloud deletes the oldest file versions until it meets the disk space limit again.

Nextcloud manages file versions using a combination of on-save pruning and scheduled cleanup. This ensures that versions are retained while respecting storage quotas.

## During Version Creation

Nextcloud automatically creates new file versions whenever a file is modified, allowing users to restore previous states when needed. After each new version is stored, the system automatically checks storage limits and retention rules. Versions are filtered according to the above pattern to

keep representative versions and remove redundant ones. If the user's quota is exceeded, auto-expiry is triggered. When storage space runs low, Nextcloud sorts all versions from oldest to newest and removes the oldest ones first, while always preserving at least the two most recent versions to free up space.

# During the Regular Background Job

Nextcloud runs a background cleanup task that automatically removes old file versions for each user. During this process, the system checks the user's version storage folder and identifies versions that are older than the configured maximum retention period or whose original files no longer exist. When an outdated or orphaned version is found, it is safely deleted from both the filesystem and the version database to reclaim storage space and maintain consistency.



Versions named by a user will never be deleted.

You may alter the default pattern in `config.php`. The default setting is `auto`, which sets the default pattern:

```
'versions_retention_obligation' => 'auto',
```

Additional options are:

- `D, auto`  
Keep versions at least for D days, apply expiration rules to all versions that are older than D days
- `auto, D`  
Delete all versions that are older than D days automatically, delete other versions according to expiration rules
- `D1, D2`  
Keep versions for at least D1 days and delete when they exceed D2 days.
- `disabled`  
Disable automatic expiration (pruning) of file versions; file versions will continue to be created, but no old file versions will be deleted automatically.

## Background job

To delete expired versions a background jobs runs every 30 minutes. It's possible to deactivate the background job and setup a (system) cron to expire the versions via occ.

Deactivate background job: `occ config:app:set --value=no files_versions background_job_expire_versions`

Activate background job: `occ config:app:delete files_versions background_job_expire_versions`

Expire versions: `occ versions:expire` or `occ versions:expire --quiet` (without the progress bar)

---

```
## #1
[] cocobook-sync [] 2026-10-28 16:28:12 UTC
[] cocobook-sync [] 2026-10-28 16:28:12 UTC
```