

FileCloud Storage Upgrade Guide

October 1, 2026

Contents

Upgrade steps	1
Troubleshooting.....	4
Repair tools	6
Rollback.....	8

Upgrade steps

The following upgrade summary lists each step of the upgrade process. Steps marked **Automatic** run without administrator input. Each step is described in detail below.

Step	Performed by	Description
1. Back up the database	Administrator	Take a full backup of the FileCloud database (MongoDB), verify that it can be restored, and retain it until you are certain you will not need to revert. Strongly recommended: a backup is the simplest and safest way to roll back.
2. Start the upgrade	Administrator	Schedule a maintenance window, then start the upgrade using the standard procedure for your system.
3. Folder structure is checked and repaired	Automatic	Before installing the new version, the upgrade checks the folder structure and repairs any problems it finds.
4. New version is installed	Automatic	The upgrade installs the 23.263.0 files.
5. Verify the upgrade	Administrator	Open several familiar folders and confirm that their files are present.

Step 1: Back up the database

Take a full backup of the FileCloud database (MongoDB), verify that it can be restored, and retain it until you are certain you will not need to revert. This step is strongly recommended: a backup is the simplest and safest way to roll back.

Follow the official backup documentation:

- Overview: [Backing Up and Restoring FileCloud Server](#)
- Linux: [FileCloud Backup and Restore - Linux Manual](#)
- Windows: [FileCloud Backup and Restore - Windows Manual](#)

- Server Backup Tool (optional): [The FileCloud Server Backup Tool](#)

The manual backup procedures stop the web server and MongoDB, then copy the MongoDB data folder (/var/lib/mongodb/ on Linux; c:\xampp\mongodb\bin\data on Windows). If you use MongoDB replica sets or host MongoDB outside the FileCloud server, back it up using your standard MongoDB procedure.

Step 2: Start the upgrade

Schedule a maintenance window, then start the upgrade using the standard procedure for your system. See [Upgrade FileCloud](#).

Maintenance window: Step 3 takes longer on installations with many folders and files. Schedule a longer maintenance window for large installations.

Step 3: Folder structure is checked and repaired

Before installing the new version, the upgrade checks the folder structure and repairs any problems it finds.

Interpretation of folder structure check and repair messages:

Message	Meaning	Action
Host `*` storage graph OK, nothing to repair	No problems were found.	None.
Host `*` storage graph repaired, upgrade not blocked	Problems were found and repaired.	Retain the log files described below.
A line beginning with Skipping host `*`	This check was completed during an earlier upgrade.	None.

In these messages, * represents the default site. On a multisite installation, the output contains a separate block for each site. Review every block.

If the check cannot run, the upgrade stops. See "The upgrade stops with an error" in the [Troubleshooting](#) section.

Log files

This step writes the following two files to the FileCloud log folder (<drive letter>\xampp\htdocs\scratch\logs on Windows; /var/www/html/scratch/logs on Linux)

File	Contents	Share with Support?
filecloud-storage-repair_<time>-<number>.log	All output displayed on screen.	Yes. This file is safe to attach.
filecloud-storage-repair_<host>_<time>-<number>.json1	A record of every change made. One file is created per site.	Yes. This file contains user and folder names, so treat it as customer data.

If a .log file exists without a corresponding .json1 file, no changes were made. If neither file exists, this step did not run.

Data safety

- **The check does not delete user data.** No files are deleted, and no folders that contain content are deleted.
- **The only items the check can delete** are empty duplicate folders (two folders with the same name in the same location, one of which is empty) and orphaned thumbnails. Thumbnails are regenerated the next time the file is viewed.

The `_recovered` folder

Because the check scans the entire folder structure, it can find files or folders that were lost before the upgrade. These items still exist in the database, but users cannot see them because their parent folder is missing or damaged. Earlier versions could leave such items behind after a failed move or delete operation.

The check does not delete these items. Instead, it moves them to a folder named `_recovered` in the owner's My Files, where the owner can see them again.

- A `_recovered` folder indicates that the upgrade **found and restored** lost items. The upgrade did not cause any data loss.
- Users can open the folder, review its contents, and move or delete items as usual.
- Every item moved to the folder is listed in the `.json1` file (see "Log files" above).

Step 4: New version is installed

The upgrade installs the 23.263.0 files.

Step 5: Verify the upgrade

Open several familiar folders and confirm that their files are present. If a file or folder appears to be missing, check the owner's `_recovered` folder (see step 3).

Impact on users: None. The web portal, apps, shared links, and API continue to work as before.

Troubleshooting

A failure does not mean data loss. If the check stops the upgrade, or the database update is interrupted, no files or folders are lost.

- If the check stops the upgrade, the current version continues to work. Any repairs completed before the stop are retained and are compatible with the current version.
- If the database update in the admin portal is interrupted, it resumes the next time the admin portal opens.
- Items that the check cannot repair are left in place. They are reported, never deleted.

Resolve the cause and run the upgrade again. Do not restore a backup solely because the upgrade stopped.

The upgrade stops with an error

The upgrade displays the following message:

```
Pre-Upgrade stopped: the storage graph repair could not be completed.
```

This message means the check could not run; it does not mean that data is damaged. The current version continues to work. The output shows `FAILED host ...` followed by the reason. Resolve the cause using the table below, then run the upgrade again. Running the upgrade again is safe.

Reason shown in the output	Resolution
Insufficient memory, or <code>memory_limit</code> could not be raised	Increase <code>memory_limit</code> in the PHP CLI <code>php.ini</code> file, or run the upgrade on a machine with more memory. The check requires approximately 640 bytes per folder plus 256 MB (for example, 1 million folders ≈ 0.9 GB).
MongoDB is unreachable, or <code>ConnectionTimeout</code>	Start MongoDB, wait until it accepts connections, and run the upgrade again.
A site is configured, but multisite is disabled	Enable multisite, or remove the site entry.
The log folder is not writable	Correct the folder permissions, or set the environment variable <code>FC_PREUPGRADE_AUDIT_DIR</code> to a folder the upgrade can write to.

If the upgrade stops again for the same reason, contact [FileCloud Support](#).

Other situations

Situation	Recommended action
The report states that items were left where they were found .	The upgrade continues. Contact FileCloud Support and provide the log files.
The report states that items share the same identifier .	Contact FileCloud Support and provide the log files.
The update page in the admin portal takes a long time.	Wait for it to finish. Long update times are expected on large sites.

Situation	Recommended action
The update page was closed, or the server restarted, during the database update.	Reopen the admin portal. The update resumes from where it stopped. Do not restore a backup.
A <code>_recovered</code> folder appears.	This is expected. See "The <code>_recovered</code> folder" under step 3 in the Upgrade steps section.
A file or folder appears to be missing after the upgrade.	Check the owner's <code>_recovered</code> folder, then run the validation tool (see the Repair tools section). Contact FileCloud Support and provide the results.

Information to provide to Support

- Both `filecloud-storage-repair_*` files
- The FileCloud version before and after the upgrade
- Whether a MongoDB backup was taken, and when
- Whether the installation is single-site or multisite, and which sites are affected
- The output of `validate_storage_integrity.php -h <host> --export <file>`
- The upgrade output and the application log

Repair tools

The repair tools are included with version 23.263.0 and are not available in earlier versions.

Location: <FileCloud web root>/resources/tools/integrity/ (for example, /var/www/html/resources/tools/integrity/ on Linux or C:\xampp\htdocs\resources\tools\integrity\ on Windows).

To run a tool: Open a terminal as an administrator, navigate to the tools folder, and run `php <tool>.php`.

Usage rules

- Without `--apply`, a tool runs in report mode and makes no changes.
- With `--apply`, a tool prompts for confirmation (`[y/N]`) before writing any changes. Only `y` continues.
- Do not suppress error output (for example, with `2>/dev/null`). The `[y/N]` prompt is written to error output; if it is suppressed, the tool appears to hang.
- Use `-h <host>` to select a single site; specify `default` for the default site. Use `--all-sites` to run against every site.
- Use `--export <file>` to write a CSV file for review. The file must not already exist. This option cannot be combined with `--all-sites`.
- Run tools with `--apply` only during a maintenance window, with user access blocked.

Available tools

Tool	Function	Makes changes?
<code>validate_storage_integrity.php</code>	Checks the folder structure.	Never
<code>recover_orphans.php</code>	Moves items with no valid parent to <code>_recovered</code> .	Only with <code>--apply</code>
<code>fix_duplicate_folders.php</code>	Deletes empty duplicate folders and renames the remaining duplicates with <code>_1</code> , <code>_2</code> , and so on.	Only with <code>--apply</code>
<code>remove_virtpath.php</code>	Used only when reverting to the previous version (see the Rollback section).	Only with <code>--apply</code>

Exit codes: 0 = clean; 1 = problems remain; 2 = error; 3 = blocked; 4 = user declined the prompt.

Note: In report mode, only `validate_storage_integrity.php` and `fix_duplicate_folders.php` return 1 when they find problems. Always review the output.

Resolving problems found by the check

1. Run the check:

```
php validate_storage_integrity.php -h <host> --export check.csv
```

2. Preview the orphan repair, then apply it:

```
php recover_orphans.php -h <host>
php recover_orphans.php -h <host> --apply
```

3. Preview the duplicate-folder repair, then apply it. Always perform this step after the orphan repair in step 2 above.

```
php fix_duplicate_folders.php -h <host> --export duplicates.csv  
php fix_duplicate_folders.php -h <host> --apply
```

4. Run the check again. It should return 0:

```
php validate_storage_integrity.php -h <host>
```

5. Re-index search. These tools do not update the search index or share paths, so review any shares on the modified folders.

If the check still reports problems, contact [FileCloud Support](#).

Rollback

The database change in version 23.263.0 is one-way: the previous version cannot run on the upgraded database as is. Select the rollback path that matches your situation:

Situation	Path
You have a valid backup and can accept losing changes made after it was taken.	A
You have no backup, or you must retain changes made after the upgrade, and version 23.263.0 is still installed.	B
The previous version's files have already been reinstalled, and FileCloud is not working.	C

For paths B and C, contact [FileCloud Support](#) before making any changes.

Path A: Restore the backup

1. Stop the FileCloud services.
2. Restore the MongoDB backup taken before the upgrade, following the restore section of the official [Linux](#) or [Windows](#) manual.
3. Install the previous FileCloud version.
4. Start the services.

All changes made after the backup was taken are lost.

Path B: Rebuild the removed data, then downgrade

Keep version 23.263.0 installed until you reach step 7 of this procedure (installing the previous version).

1. Check whether the data was removed:

```
php remove_virtpath.php -h <host>
```

If the output shows `Folders affected : 0`, the data was removed; continue with step 2 below. If the value is greater than 0, the data was never removed; skip to step 7 below.

2. Stop cron, the orchestrator, and Solr, and block user access. Leave the web server and MongoDB running.
3. Check and repair the folder structure by completing steps 1–4 of "Resolving problems found by the check" in the [Repair tools](#) section.
4. Preview the rebuild, then run it:

```
php remove_virtpath.php -h <host> --rollback
php remove_virtpath.php -h <host> --rollback --apply
```

Confirm that the output shows `Applied` and `YES`, and that the `Note` field is empty. If `Note` contains a value, contact [FileCloud Support](#).

5. Run the check again:

```
php validate_storage_integrity.php -h <host>
```

6. Repeat steps 1–5 of this procedure for each site.
7. Install the previous FileCloud version.

8. Start the services and re-index search.

Important: Rebuild the data before downgrading. If you later upgrade to version 23.263.0 again, the upgrade removes the data again.

Path C: The previous version is reinstalled and FileCloud is not working

1. Reinstall version 23.263.0. No database update runs, so this step only restores the program files.
2. Follow path B.