

Conflicts



- [Overview](#)
 - [Example](#)
 - [Uploading conflicts \(experimental\)](#)
-

Overview

The Nextcloud desktop client uploads local changes and downloads remote changes. When a file has changed on the local side and on the remote between synchronization runs the client will be unable to resolve the situation on its own. It will create a conflict file with the local version, download the remote version and notify the user that a conflict occurred which needs attention.

Example

Imagine there is a file called `mydata.txt` in your synchronized folder. It has not changed for a while and contains the text "contents" locally and remotely. Now, nearly at the same time you update it locally to say "local contents" while the file on the server gets updated to contain "remote contents" by someone else.

When attempting to upload your local changes the desktop client will notice that the server version has also changed. It creates a conflict and you will now have two files on your local machine:

- `mydata.txt` containing "remote contents"
- `mydata (conflicted copy 2018-04-10 093612).txt` containing "local contents"

In this situation the file `mydata.txt` has the remote changes (and will continue to be updated with further remote changes when they happen), but your local adjustments have not been sent to the server (unless the server enables conflict uploading, see below).

The desktop client notifies you of this situation via system notifications, the system tray icon and a yellow "unresolved conflicts" badge in the account settings window. Clicking this badge shows a list that includes the unresolved conflicts and clicking one of them opens an explorer window pointing

at the relevant file.

To resolve this conflict, open both files, compare the differences and copy your local changes from the "conflicted copy" file into the base file where applicable. In this example you might change `mydata.txt` to say "local and remote contents" and delete the file with "conflicted copy" in its name. With that, the conflict is resolved.

Uploading conflicts (experimental)

By default the conflict file (the file with "conflicted copy" in its name that contains your local conflicting changes) is not uploaded to the server. The idea is that you, the author of the changes, are the best person for resolving the conflict and showing the conflict to other users might create confusion.

However, in some scenarios it makes a lot of sense to upload these conflicting changes such that local work can become visible even if the conflict won't be resolved immediately.

In the future there might be a server-wide switch for this behavior. For now it can already be tested by setting the environment variable `OWNCLOUD_UPLOAD_CONFLICT_FILES=1`.

📁 #1
📁 cocobook-sync 📁 2026-10-28 16:23:49 UTC
📁 cocobook-sync 📁 2026-10-28 16:23:49 UTC