

Troubleshooting



- [Identifying Basic Functionality Problems](#)
 - ["CSync unknown error"](#)
 - ["Connection closed" message when syncing files](#)
 - [Connection issues with the macOS client on "insecure" connections](#)
 - [Isolating other issues](#)
 - [Log Files](#)
 - [Obtaining the Client Log File](#)
 - [Create Debug Archive](#)
 - [Keyboard shortcut](#)
 - [Command line](#)
 - [Config file](#)
 - [Nextcloud server Log File](#)
 - [Webserver Log Files](#)
 - [Core Dumps](#)
 - [Citrix Workspace known issues](#)
-

The following two general issues can result in failed synchronization:

- The server setup is incorrect.
- The client contains a bug.

When reporting bugs, it is helpful if you first determine what part of the system is causing the issue.

Identifying Basic Functionality Problems

Performing a general Nextcloud Server test

The first step in troubleshooting synchronization issues is to verify that you can log on to the Nextcloud web application. To verify connectivity to the Nextcloud server try logging in via your Web browser.

If you are not prompted for your username and password, or if a red warning box appears on the page, your server setup requires modification. Please verify that your server installation is working correctly.

Ensure the WebDAV API is working

If all desktop clients fail to connect to the Nextcloud Server, but access using the Web interface functions properly, the problem is often a misconfiguration of the WebDAV API.

The Nextcloud Client uses the built-in WebDAV access of the server content. Verify that you can log on to Nextcloud's WebDAV server. To verify connectivity with the Nextcloud WebDAV server:

- Open a browser window and enter the address to the Nextcloud WebDAV server.

For example, if your Nextcloud instance is installed at `http://yourserver.com/nextcloud`, your WebDAV server address is `http://yourserver.com/nextcloud/remote.php/dav`.

If you are prompted for your username and password but, after providing the correct credentials, authentication fails, please ensure that your authentication backend is configured properly.

Use a WebDAV command line tool to test

A more sophisticated test method for troubleshooting synchronization issues is to use a WebDAV command line client and log into the Nextcloud WebDAV server. One such command line client -- called `cadaver` -- is available for Linux distributions. You can use this application to further verify that the WebDAV server is running properly using PROPFIND calls.

As an example, after installing the `cadaver` app, you can issue the `propget` command to obtain various properties pertaining to the current directory and also verify WebDAV server connection.

"CSync unknown error"

If you see this error message stop your client, delete the `.sync_XXXXXX.db` file, and then restart your client. There is a hidden `.sync_XXXXXX.db` file inside the folder of every account configured on your client.

“ ”

Please note that this will also erase some of your settings about which files to download.

See <https://github.com/owncloud/client/issues/5226> for more discussion of this issue.

"Connection closed" message when syncing files

This message can be caused by using chunks that are too big or time-outs that are set too liberally. You can configure the chunking behavior of the client in the config file. For example, change these settings:

<code>chunkSize</code>	<code>10000000</code> (10 MB)	Specifies the chunk size of uploaded files in bytes. The client will dynamically adjust this size within the maximum and minimum bounds (see below).
<code>minChunkSize</code>	<code>1000000</code> (1 MB)	Specifies the minimum chunk size of uploaded files in bytes.
<code>maxChunkSize</code>	<code>50000000</code> (1000 MB)	Specifies the maximum chunk size of uploaded files in bytes.
<code>targetChunkUploadDuration</code>	<code>6000</code> (1 minute)	Target duration in milliseconds for chunk uploads. The client adjusts the chunk size until each chunk upload takes approximately this long. Set to 0 to disable dynamic chunk sizing.

Setting `maxChunkSize` to 50000000, for example, will decrease the individual chunk to about 50 mb. This causes additional overhead but might be required in some situations, for example behind CloudFlare which has been seen limiting upload chunks to 100mb. In other situations, limiting `targetChunkUploadDuration` can help to avoid time-outs.

Connection issues with the macOS client on "insecure" connections

When using macOS devices to connect to a Nextcloud server that uses a what maybe be classified as an insecure connection (i.e. connecting to a server with a self-signed certificate, or a certificate with what Apple may consider an insufficiently secure cipher), the macOS client may not connect to the server. This is because macOS requires a valid certificate to establish a connection.

To resolve this issue, you must ensure the server is signed with a certificate that is accepted by Apple's App Transport Security requirements. More information on the requirements can be found in Apple's documentation pages.

<https://developer.apple.com/documentation/security/preventing-insecure-network-connections>

Isolating other issues

Other issues can affect synchronization of your Nextcloud files:

- If you find that the results of the synchronizations are unreliable, please ensure that the folder to which you are synchronizing is not shared with other synchronization applications.
- Synchronizing the same directory with Nextcloud and other synchronization software such as Unison, rsync, Microsoft Windows Offline Folders, or other cloud services such as Dropbox or Microsoft SkyDrive is not supported and should not be attempted. In the worst case, it is possible that synchronizing folders or files using Nextcloud and other synchronization software or services can result in data loss.
- If you find that only specific files are not synchronized, the synchronization protocol might be having an effect. Some files are automatically ignored because they are system files, other files might be ignored because their filename contains characters that are not supported on certain file systems. For more information about ignored files, see ignored-files.
- If you are operating your own server, and use the local storage backend (the default), make sure that Nextcloud has exclusive access to the directory.



The data directory on the server is exclusive to Nextcloud and must not be modified manually.

- If you are using a different file backend on the server, you can try to exclude a bug in the backend by reverting to the built-in backend.
- If you are experiencing slow upload/download speed or similar performance issues be aware that those could be caused by on-access virus scanning solutions, either on the server (like the files_antivirus app) or the client.

Log Files

Effectively debugging software requires as much relevant information as can be obtained. To assist the Nextcloud support personnel, please try to provide as many relevant logs as possible. Log output can help with tracking down problems and, if you report a bug, log output can help to resolve an issue more quickly.

“

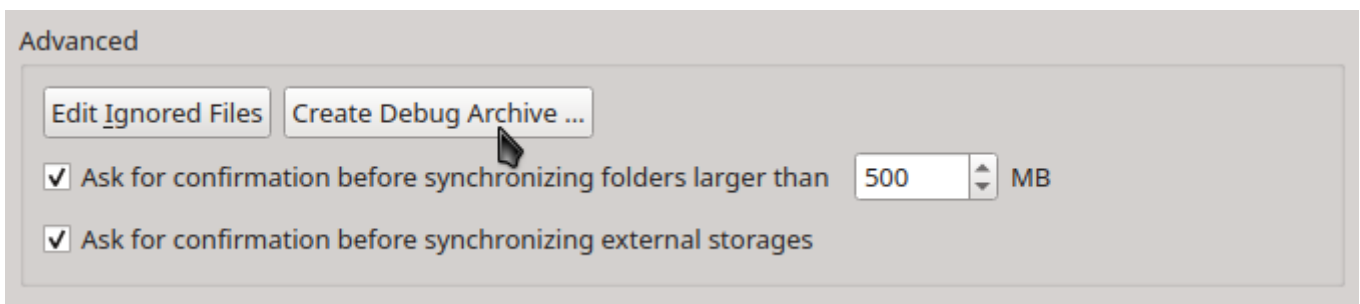
“ □

Log files contain sensitive information. You may wish to redact sensitive details or to only share limited excerpts.

Obtaining the Client Log File

Create Debug Archive

Since the 3.1.0 release we made it easier for users to provide debug information: debug logging is enabled by default with expiration time set to 24 hours and under the "General" settings, you can click on "Create Debug Archive ..." to pick the location of where the desktop client will export the logs and the database to a zip file.

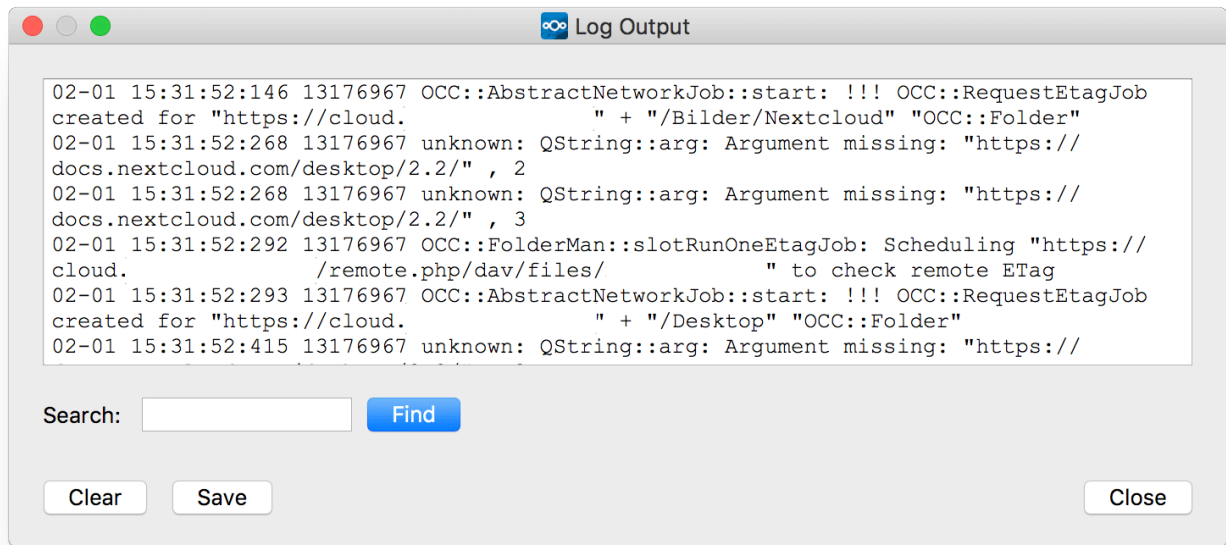


Keyboard shortcut

Another way to obtain the client log file:

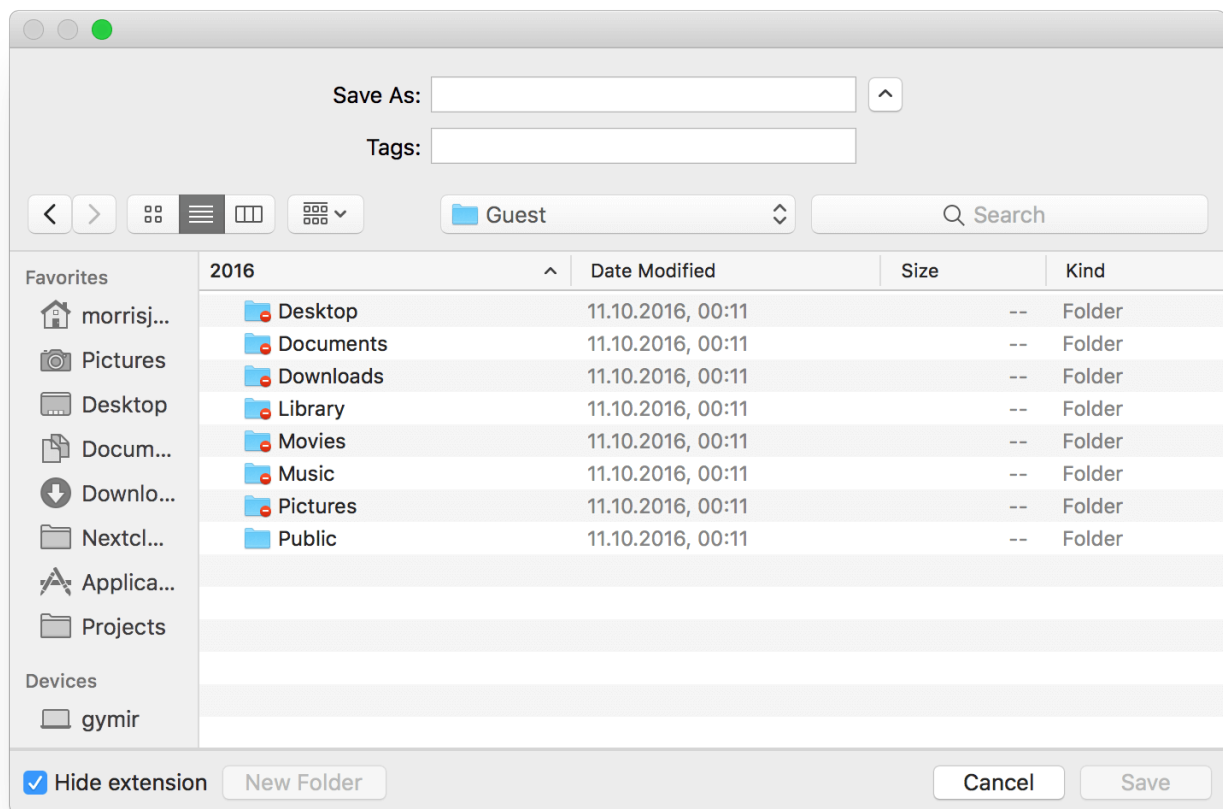
1. Open the Nextcloud Desktop Client.
2. Press **F12** or **Ctrl + L** on your keyboard.

The Log Output window opens.



3. Click the 'Save' button.

The Save Log File window opens.



4. Navigate to a location on your system where you want to save your log file.
5. Name the log file and click the 'Save' button.

“ The log file is saved in the location specified.

Command line

Alternatively, you can launch the Nextcloud Log Output window using the `--logwindow` command. After issuing this command, the Log Output window opens to show the current log. You can then follow the same procedures mentioned above to save the log to a file.

“ “

You can also open a log window for an already running session, by restarting the client using the following command:

- Windows: `C:\Program Files\Nextcloud\nextcloud.exe --logwindow`
- macOS: `/Applications/nextcloud.app/Contents/MacOS/nextcloud --logwindow`
- Linux: `nextcloud --logwindow`

Config file

The Nextcloud client enables you to save log files directly to a predefined file or directory. This is a useful option for troubleshooting sporadic issues as it enables you to log large amounts of data and bypass the limited buffer settings associated with the log window.

To enable logging to a directory, stop the client and add the following to the General section in the configuration file:

```
[General]
logDebug=true
logExpire=<hours>
logDir=<dir>
```

Independent of platform you must use slash (/) as a path separator:

“

“ □

* Correct: C:/Temp * Not correct: C:Temp

As an example, to keep log data for two days in a directory called temp:

```
[General]
logDebug=true
logExpire=48
logDir=C:/Temp
```

Once you restart the client, you will find the log file in the `<dir>` defined in `logDir`.

“

“ □

You will find the configuration file in the following locations:

- Microsoft Windows systems:
`%APPDATA%\Nextcloud\nextcloud.cfg`
- macOS systems:
`$HOME/Library/Preferences/Nextcloud/nexcloud.cfg`
- Linux distributions:
`$HOME/.config/Nextcloud/nexcloud.cfg`

Alternatively, you can start the client in the command line with parameters:

1. To save to a file, start the client using the `--logfile <file>` command, where `<file>` is the filename to which you want to save the file.
2. To save to a directory, start the client using the `--logdir <dir>` command, where `<dir>` is an existing directory.

When using the `--logdir` command, each sync run creates a new file. To limit the amount of data that accumulates over time, you can specify the `--logexpire <hours>` command. When combined with the `--logdir` command, the client automatically erases saved log data in the directory that is older than the specified number of hours.

As an example, to define a test where you keep log data for two days, you can issue the following command:

```
` nextcloud --logdir /tmp/nextcloud_logs --logexpire 48 `
```

Nextcloud server Log File

The Nextcloud server also maintains an Nextcloud specific log file. This log file must be enabled through the Nextcloud Administration page. On that page, you can adjust the log level. We recommend that when setting the log file level that you set it to a verbose level like `Debug` or `Info`.

You can view the server log file using the web interface or you can open it directly from the file system in the Nextcloud server data directory.

See [Logging](#) in the admin manual for details on configuring log levels and log file locations.

Webserver Log Files

It can be helpful to view your webserver's error log file to isolate any Nextcloud-related problems. For Apache on Linux, the error logs are typically located in the `/var/log/apache2` directory. Some helpful files include the following:

- `error_log` -- Maintains errors associated with PHP code.
- `access_log` -- Typically records all requests handled by the server; very useful as a debugging tool because the log line contains information specific to each request and its result.

You can find more information about Apache logging at

`http://httpd.apache.org/docs/current/logs.html`.

Core Dumps

On macOS and Linux systems, and in the unlikely event the client software crashes, the client is able to write a core dump file. Obtaining a core dump file can assist Nextcloud Customer Support tremendously in the debugging process.

To enable the writing of core dump files, you must define the `OWNCLOUD_CORE_DUMP` environment variable on the system.

For example:

```
` OWCLOUD_CORE_DUMP=1 nextcloud `
```

This command starts the client with core dumping enabled and saves the files in the current working directory.



Core dump files can be fairly large. Before enabling core dumps on your system, ensure that you have enough disk space to accommodate these files. Also, due to their size, we strongly recommend that you properly compress any core dump files prior to sending them to Nextcloud Customer Support.

Citrix Workspace known issues

These are known issues when running the desktop client in a Citrix workspace:

- The Windows user's Roaming profile needs to be persisted between sessions. The failure on doing so will result in users having to set up their account again in every new session.
- The user's synchronization folder also needs to be persisted between sessions. The client will throw errors because it can not find the sync folder once the users logs in a new session.
- Every time the user logs in a Citrix environment, it creates a session with the desktop client and in that session the client will sync the user's files, that can quickly lead to the storage running out of space.

🔍 #3

🔍 cocobook-sync 🔍 2026-10-28 16:29:41 UTC

🔍 cocobook-sync 🔍 2026-10-29 01:09:24 UTC