

# Reducing program lag

## hydrus is cpu and hdd hungry

The hydrus client manages a lot of complicated data and gives you a lot of power over it. To add millions of files and tags to its database, and then to perform difficult searches over that information, it needs to use a lot of CPU time and hard drive time--sometimes in small laggy blips, and occasionally in big 100% CPU chunks. I don't put training wheels or limiters on the software either, so if you search for 300,000 files, the client will try to fetch that many.

In general, the client works best on snappy computers with low-latency hard drives where it does not have to constantly compete with other CPU- or HDD- heavy programs. Running hydrus on your games computer is no problem at all, but if you leave the client on all the time, then make sure under the options it is set not to do idle work while your CPU is busy, so your games can run freely. Similarly, if you run two clients on the same computer, you should have them set to work at different times, because if they both try to process 500,000 tags at once on the same hard drive, they will each slow to a crawl.

If you run on an HDD, keeping it defragged is very important, and good practice for all your programs anyway. Make sure you know what this is and that you do it.

## maintenance and processing

I have attempted to offload most of the background maintenance of the client (which typically means repository processing and internal database defragging) to time when you are not using the client. This can either be 'idle time' or 'shutdown time'. The calculations for what these exactly mean are customisable in *file->options->maintenance and processing*.

If you run a quick computer, you likely don't have to change any of these options. Repositories will synchronise and the database will stay fairly optimal without you even noticing the work that is going on. This is especially true if you leave your client on all the time.

If you have an old, slower computer though, or if your hard drive is high latency, make sure these options are set for whatever is best for your situation. Turning off idle time completely is often helpful as some older computers are slow to even recognise--mid task--that you want to use the client again, or take too long to abandon a big task half way through. If you set your client to only do work on shutdown, then you can control exactly when that happens.

# reducing search and general gui lag

Searching for tags via the autocomplete dropdown and searching for files in general can sometimes take a very long time. It depends on many things. In general, the more predicates (tags and system:something) you have active for a search, and the more specific they are, the faster it will be.

You can also look at *file->options->speed and memory*, again especially if you have a slow computer. Increasing the autocomplete thresholds is very often helpful. You can even force autocompletes to only fetch results when you manually ask for them.

Having lots of thumbnails open or downloads running can slow many things down. Check the 'pages' menu to see your current session weight. If it is about 50,000, or you have individual pages with more than 10,000 files or download URLs, try cutting down a bit.

## finally - profiles

Lots of my code remains unoptimised for certain situations. My development environment only has a few thousand images and a few million tags. As I write code, I am usually more concerned with getting it to work at all rather than getting it to work fast for every possible scenario. So, if something is running slow for you, but your computer is otherwise working fine, let me know and I can almost always speed it up.

Let me know:

- The general steps to reproduce the problem (e.g. "Running system:numtags>4 is ridiculously slow on its own on 'all known tags'.")
- Your operating system and its version (e.g. "Windows 8.1")
- Your computer's general power (e.g. "A couple of years old. It runs most stuff ok.")
- The type of hard drive you are running hydrus from. (e.g. "A 2TB 7200rpm drive that is 20% full. I regularly defrag it.")
- Any *profiles* you have collected.

A *profile* is a large block of debug text that lets me know which parts of my code are running slow for you. A profile for a single call looks like [this](#).

It is very helpful to me to have a profile. You can generate one by going *help->debug->xxx profile mode*, which tells the client to generate profile information for every subsequent xxx request. This can be spammy, so don't leave it on for a very long time (you can turn it off by hitting the help menu entry again).

For most problems, you probably want *db profile mode*.

Turn on a profile mode, do the thing that runs slow for you (importing a file, fetching some tags, whatever), and then check your database folder (most likely *install\_dir/db*) for a new 'client profile - DATE.log' file. This file will be filled with several sets of tables with timing information. Please send that whole file to me, or if it is too large, cut what seems important. It should not contain any personal information, but feel free to look through it.

There are several ways to [contact me](#).

---

Revision #1

Created 12 March 2021 16:39:47 by CuddleBear

Updated 12 March 2021 18:24:07 by CuddleBear