

Starting out

- [Introduction](#)
- [Getting Started: Installing](#)
- [Getting started: Files](#)
- [Getting started: Tags](#)
- [Getting started: Downloading](#)
- [Getting started: Ratings](#)
- [Access keys](#)

Introduction

on being anonymous

Nearly all sites use the same pseudonymous username/password system, and nearly all of them have the same drama, sockpuppets, and egotistical mods. Censorship is routine. That works for many people, but not for me.

I enjoy being anonymous online. When you aren't afraid of repercussions, you can be as truthful as you want. You can have conversations that can happen nowhere else. It's fun!

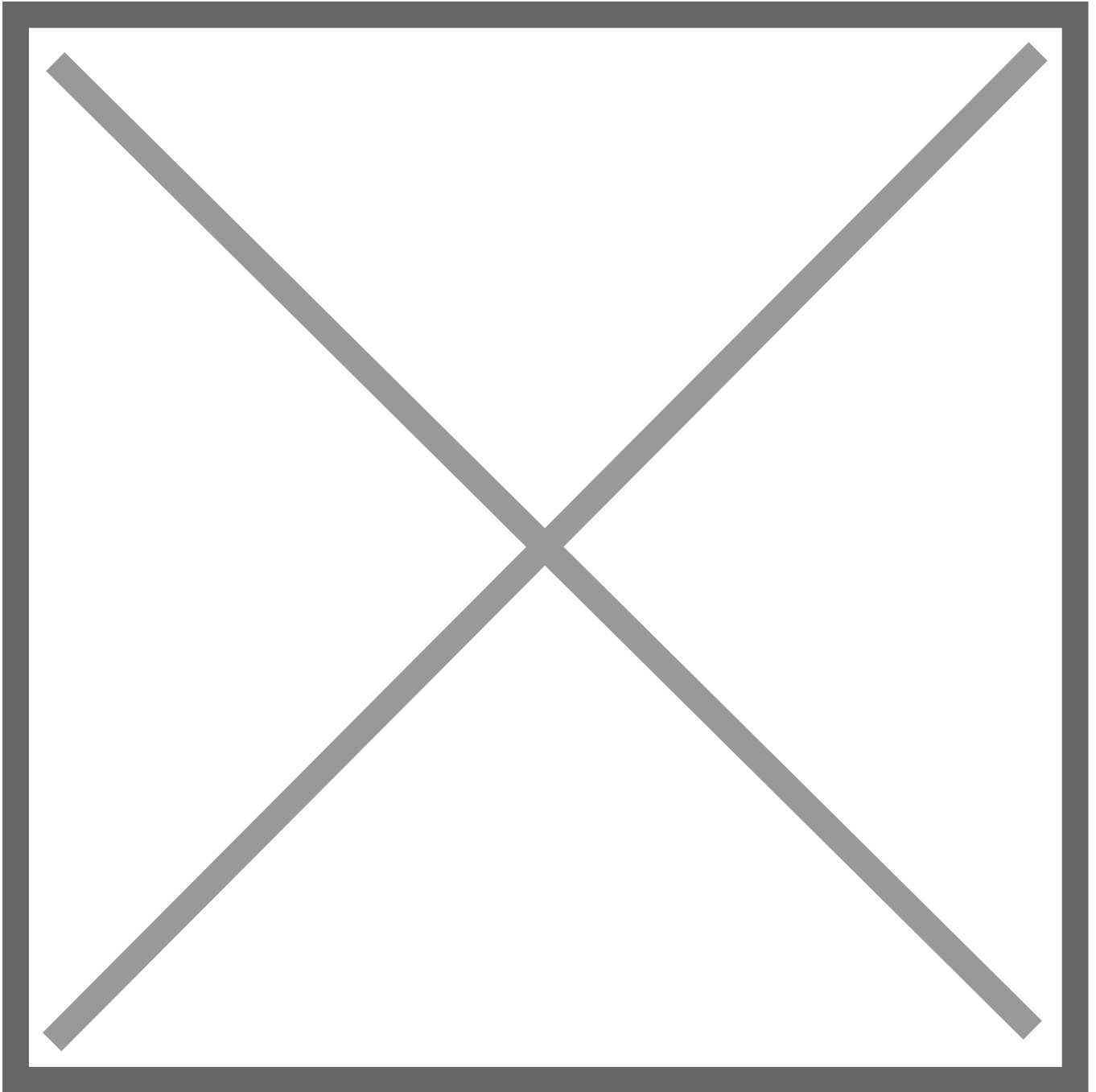
I've been on the imageboards for a long time, saving everything I like to my hard drive. After a while, the whole collection was just too large to manage on my own.

the hydrus network

So! I'm developing a program that helps people organise their files together anonymously. I want to help you do what you want with your stuff, and that's it. You can share some tags and files with other people if you want to, but you don't have to connect to anything if you don't. **The default is complete privacy, no sharing**, and every upload requires a conscious action on your part. I don't plan to ever record metrics on users, nor serve ads, nor charge for my software. The software never phones home.

This does a lot more than a normal image viewer. If you are totally new to the idea of personal media collections and tagging, I suggest you start slow, walk through the getting started guides, and experiment doing different things. If you aren't sure on what a button does, try clicking it! You'll be importing thousands of files and applying *tens* of thousands of tags in no time.

The client is chiefly a file database. It stores your files inside its own folders, managing them far better than an explorer window or some online gallery. Here's a screenshot of one of my test installs with a search showing all files:



As well as the client, there is also a server that anyone can run to store files or tags for sharing between many users. The mechanics of running a server is usually confusing to new users, so wait a little while before you explore this. Some users run a public tag repository with hundreds of millions of tags that you can access and contribute to if you wish.

I have many plans to expand the client and the network.

statement of principles

- No speech should be outlawed.
- Everyone should be able to control their own media diet.

- Computer data and network logs should be absolutely private.

None of the above are currently true, but I would love to live in a world where they were. My software is an attempt to move us a little closer.

I try to side with the person over the authority, the distributed over the centralised. I still use gmail and youtube just like pretty much everyone, but I would rather be using different systems, especially in ten years. No one seemed to be making what I wanted for file management, so I decided to do it myself, and here we are.

If, after a few months, you find you enjoy the software and would like to further support it, I have set up a simple no-reward patreon, which you can read more about [here](#).

license

These programs are free software. Everything I, hydrus dev, have made is under the Do What The Fuck You Want To Public License, Version 3, as published by Kris Craig. See <https://github.com/sirkris/WTFPL/blob/master/WTFPL.md> for more details.

Do what the fuck you want to with my software, and if shit breaks, DEAL WITH IT.

Getting Started: Installing

If any of this is confusing, a simpler guide is [here](#), and some video guides are [here](#)!

downloading

You can get the latest release at [my github releases page](#).

I try to release a new version every Wednesday by 8pm EST and write an accompanying post on [my tumblr](#) and a Hydrus Network General thread on [8chan.moe /t/](#).

installing

The hydrus releases are 64-bit only. If you are a python expert, there is the slimmest chance you'll be able to get it running from source on a 32-bit machine, but it would be easier just to find a newer computer to run it on.

for Windows:

- If you want the easy solution, download the .exe installer. Run it, hit ok several times.
- If you know what you are doing and want a little more control, get the .zip. Don't extract it to Program Files unless you are willing to run it as administrator every time (it stores all its user data inside its own folder). You probably want something like D:\hydrus.
- *Note if you run <Win10, you may need [Visual C++ Redistributable for Visual Studio 2015](#), if you don't already have it for vidya. If you run Win7, you will need some/all core OS updates released before 2017.*
- If you use Windows 10 N (a version of Windows without some media playback features), you will likely need the 'Media Feature Pack'. There have been several versions of this, so it may best found by searching for the latest version or hitting Windows Update, but otherwise check [here](#).

for macOS:

- Get the .dmg App. Open it, drag it to Applications, and check the readme inside.

for Linux:

- Get the .tag.gz. Extract it somewhere useful and create shortcuts to 'client' and 'server' as you like. I build on Ubuntu, so if you run something else, compatibility is hit and miss.

- If you use Arch Linux, you can check out the AUR package a user maintains [here](#).
- If you have problems running the Ubuntu build, users with some python experience generally find running from source works well.
- You might need to get 'libmpv1' to get mpv working and playing video/audio. This is the mpv library, not the player. Check help->about to see if it is available--if not, see if you can get it with *apt*.
- You can also try [running the Windows version in wine](#).

from source:

- If you have some python experience, you can [run from source](#).

Hydrus stores all its data—options, files, subscriptions, *everything*—entirely inside its own directory. You can extract it to a usb stick, move it from one place to another, have multiple installs for multiple purposes, wrap it all up inside a truecrypt volume, whatever you like. The .exe installer writes some unavoidable uninstall registry stuff to Windows, but the 'installed' client itself will run fine if you manually move it.

However, for macOS users: the Hydrus App is **non-portable** and puts your database in ~/Library/Hydrus (i.e. /Users/[You]/Library/Hydrus). You can update simply by replacing the old App with the new, but if you wish to backup, you should be looking at ~/Library/Hydrus, not the App itself.

updating

Hydrus is imageboard-tier software, wild and fun but unprofessional. It is written by one Anon spinning a lot of plates. Mistakes happen from time to time, usually in the update process. There are also no training wheels to stop you from accidentally overwriting your whole db if you screw around. Be careful when updating. Make backups beforehand!

Hydrus does not auto-update. It will stay the same version unless you download and install a new one.

Although I put out a new version every week, you can update far less often if you want. The client keeps to itself, so if it does exactly what you want and a new version does nothing you care about, you can just leave it. Other users enjoy updating every week, simply because it makes for a nice schedule. Others like to stay a week or two behind what is current, just in case I mess up and cause a temporary bug in something they like.

A user has written a longer and more formal guide to updating, and information on the 334->335 step [here](#).

The update process:

- If the client is running, close it!
- If you maintain a backup, run it now!
- If you use the installer, just download the new installer and run it. It should detect where the last install was and overwrite everything automatically.
- If you extract, then just extract the new version right on top of your current install and overwrite manually.
- Start your client or server. It may take a few minutes to update its database. I will say in the release post if it is likely to take longer.

Unless the update specifically disables or reconfigures something, all your files and tags and settings will be remembered after the update.

Releases typically need to update your database to their version. New releases can retroactively perform older database updates, so if the new version is v255 but your database is on v250, you generally only need to get the v255 release, and it'll do all the intervening v250->v251, v251->v252, etc... update steps in order as soon as you boot it. If you need to update from a release more than, say, ten versions older than current, see below. You might also like to skim the release posts or [changelog](#) to see what is new.

Clients and servers of different versions can usually connect to one another, but from time to time, I make a change to the network protocol, and you will get polite error messages if you try to connect to a newer server with an older client or *vice versa*. There is still no *need* to update the client--it'll still do local stuff like searching for files completely fine. Read my release posts and judge for yourself what you want to do.

clean installs

This is only relevant if you update and cannot boot at all.

Very rarely, hydrus needs a clean install. This can be due to a special update like when we moved from 32-bit to 64-bit or needing to otherwise 'reset' a custom install situation. The problem is usually that a library file has been renamed in a new version and hydrus has trouble figuring out whether to use the older one (from a previous version) or the newer.

In any case, if you cannot boot hydrus and it either fails silently or you get a crash log or system-level error popup complaining in a technical way about not being able to load a dll/pyd/so file, you may need a clean install, which essentially means clearing any old files out and reinstalling.

However, you need to be careful not to delete your database! It sounds silly, but at least one user has made a mistake here. The process is simple, do not deviate:

- Make a backup if you can!
- Go to your install directory.

- Delete all the files and folders except the 'db' dir (and all of its contents, obviously).
- Reinstall/extract hydrus as you normally do.

After that, you'll have a 'clean' version of hydrus that only has the latest version's dlls. If hydrus still will not boot, I recommend you roll back to your last working backup and let me, hydrus dev, know what your error is.

big updates

If you have not updated in some time--say twenty versions or more--doing it all in one jump, like v250->v290, is likely not going to work. I am doing a lot of unusual stuff with hydrus, change my code at a fast pace, and do not have a ton of testing in place. Hydrus update code often falls to bitrot, and so some underlying truth I assumed for the v255->v256 code may not still apply six months later. If you try to update more than 50 versions at once (i.e. trying to perform more than a year of updates in one go), the client will give you a polite error rather than even try.

As a result, if you get a failure on trying to do a big update, try cutting the distance in half--try v270 first, and then if that works, try v270->v290. If it doesn't, try v260, and so on.

If you narrow the gap down to just one version and still get an error, please let me know. I am very interested in these sorts of problems and will be happy to help figure out a fix with you (and everyone else who might be affected).

backing up

Maintaining a regular backup is important for hydrus. The program stores a lot of complicated data that you will put hours and hours of work into, and if you only have one copy and your hard drive breaks, you could lose everything. This has happened before, and it sucks to go through. Don't let it be you.

If you do not already have a backup routine for your files, this is a great time to start. I now run a backup every week of all my data so that if my computer blows up or anything else awful happens, I'll at worst have lost a few days' work. Before I did this, I once lost an entire drive with tens of thousands of files, and it felt awful. If you are new to saving a lot of media, I hope you can avoid what I felt. ;_;

I use ToDoList to remind me of my jobs for the day, including backup tasks, and FreeFileSync to actually mirror over to an external usb drive. I recommend both highly (and for ToDoList, I recommend hiding the complicated columns, stripping it down to a simple interface). It isn't a huge expense to get a couple-TB usb drive either--it is **absolutely** worth it for the peace of mind.

By default, hydrus stores all your user data in one location, so backing up is simple:

• the simple way - inside the client

Go *database->set up a database backup location* in the client. This will tell the client where you want your backup to be stored. A fresh, empty directory on a different drive is ideal.

Once you have your location set up, you can thereafter hit *database->update database backup*. It will lock everything and mirror your files, showing its progress in a popup message. The first time you make this backup, it may take a little while (as it will have to fully copy your database and all its files), but after that, it will only have to copy new or altered files and should only ever take a couple of minutes.

Advanced users who have migrated their database across multiple locations will not have this option--use an external program in this case.

• the powerful way - using an external program

If you would like to integrate hydrus into a broader backup scheme you already run, or you are an advanced user with a complicated hydrus install that you have migrated across multiple drives, then you need to backup two things: the *client*.db* files and your *client_files* directory(ies). By default, they are all stored in *install_dir/db*. The *.db* files contain your settings and file metadata like inbox/archive and tags, while the *client_files* subdirs store your actual media and its thumbnails. If everything is still under *install_dir/db*, then it is usually easiest to just backup the whole install dir, keeping a functional 'portable' copy of your install that you can restore no prob. Make sure you keep the *.db* files together--they are not interchangeable and mostly useless on their own! Shut the client down while you run the backup, obviously.

Do not put your live database in a folder that continuously syncs to a cloud backup. Many of these services will interfere with a running client and can cause database corruption. If you still want to use a system like this, either turn the sync off while the client is running, or use the above backup workflows to safely backup your client to a separate folder that syncs to the cloud.

I recommend you always backup before you update, just in case there is a problem with my code that breaks your database. If that happens, please [contact me](#), describing the problem, and revert to the functioning older version. I'll get on any problems like that immediately.

Getting started: Files

If any of this is confusing, a simpler guide is [here](#), and some video guides are [here](#)!

a warning

Hydrus can be powerful, and you control everything. By default, you are not connected to any servers and absolutely nothing is shared with other users--and you can't accidentally one-click your way to exposing your whole collection--but if you tag private files with real names and click to upload that data to a tag repository that other people have access to, the program won't try to stop you. If you want to do private sexy slideshows of your shy wife, that's great, but think twice before you upload files or tags anywhere, particularly as you learn. It is **impossible** to contain leaks of private information.

There are no limits and few brakes on your behaviour. It is possible to import millions of files. For many new users, their first mistake is downloading too much too fast in overexcitement and becoming overwhelmed. Take things slow and figure out good processing workflows that work for your schedule before you start adding 500 subscriptions.

the problem

If you have ever seen something like this--



--then you already know the problem: using a filesystem to manage a lot of images sucks.

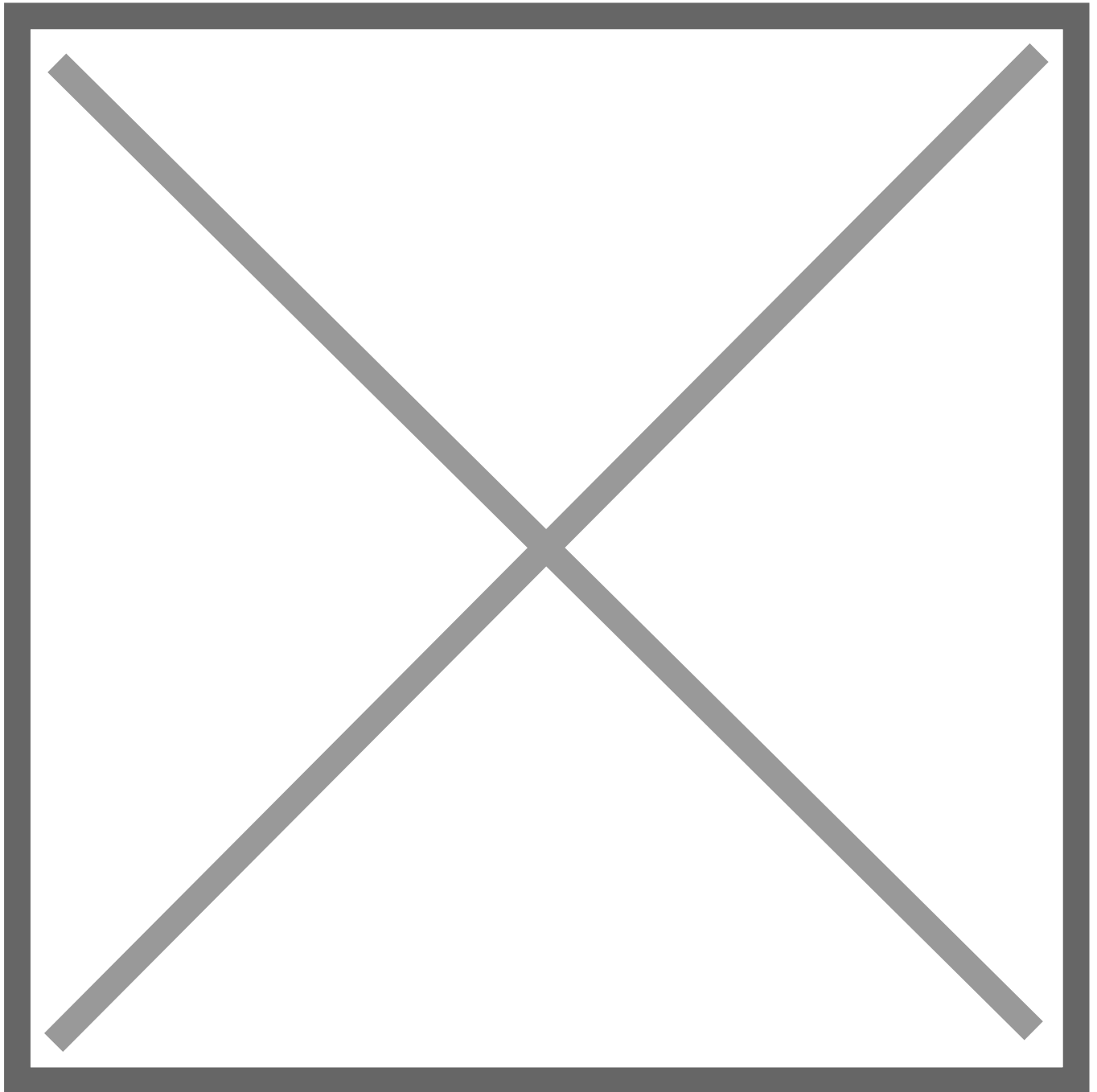
Finding the right picture quickly can be difficult. Finding everything by a particular artist at a particular resolution is unthinkable. Integrating new files into the whole nested-folder mess is a further pain, and most operating systems bug out when displaying 10,000+ thumbnails.

so, what does the hydrus client do?

Let's first focus on *importing* files.

When you first boot the client, you will see a blank page. There are no files in the database and so there is nothing to search. To get started, I suggest you simply drag-and-drop a folder with a

hundred or so images onto the main window. A dialog will appear affirming what you want to import. Ok that, and a new page will open. Thumbnails will stream in as the software processes each file.

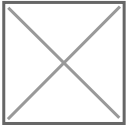


The files are being imported into the client's database. The client discards their filenames.

Notice your original folder and its files are untouched. You can move the originals somewhere else, delete them, and the client will still return searches fine. In the same way, you can delete from the client, and the original files will remain unchanged--import is a **copy**, not a move, operation. The client performs all its operations on its internal database, which holds copies of the files it imports. If you find yourself enjoying using the client and decide to completely switch over, you can delete the original files you import without worry. You can always export them back again later.

Now:

- Click on a thumbnail; it'll show in the preview screen, bottom left.
- Double- or middle-click the thumbnail to open the media viewer. You can hit 'f' to switch between giving the fullscreen a frame or not. You can use your scrollwheel or page up/down to browse the media and ctrl+scrollwheel to zoom in and out.
- Move your mouse to the top-left, top-middle and top-right of the media viewer. You should see some 'hover' panels pop into place.



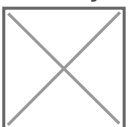
The one on the left is for tags, the middle is for browsing and zoom commands, and the right is for status and ratings icons. You will learn more about these things as you get more experience with the program.

- Press enter/return or double/middle-click again to close the media viewer.
- You can quickly select multiple files by shift- or ctrl- clicking. Notice how the status bar at the bottom of the screen updates with the number selected and their total size. Right-clicking your selection will present another summary and many actions.
- Hit F9 to bring up a new page chooser. You can navigate it with the arrow keys, your numpad, or your mouse.
- On the left of a normal search page is a text box. When it is focused, a dropdown window appears. It looks like this:



This is where you enter the predicates that define the current search. If the text box is empty, the dropdown will show 'system' tags that let you search by file metadata such as file size or animation duration. To select one, press the up or down arrow keys and then enter, or double click with the mouse.

When you have some tags in your database, typing in the text box will search them:



The (number) shows how many files have that tag, and hence how large the search result will be if you select that tag.

Clicking 'searching immediately' will pause the searcher, letting you add several tags in a row without sending it off to get results immediately. Ignore the other buttons for now-- you will figure them out as you gain experience with the program.

- You can remove from the list of 'active tags' in the box above with a double-click, or by entering the exact same tag again through the dropdown.
- Play with the system tags more if you like, and the sort-by dropdown. The collect-by dropdown is advanced, so wait until you understand *namespaces* before expecting it to do anything.
- To close a page, middle-click its tab.

The client can currently import the following mimetypes:

- **image/bmp** (.bmp - converted to image/png on import)
- **image/gif** (.gif)
- **image/png** (.png)
- **image/apng** (.apng)
- **image/jpeg** (.jpg)
- **image/tiff** (.tiff)
- **image/webp** (.webp)
- **video/x-msvideo** (.avi)
- **video/x-flv** (.flv)
- **video/x-matroska** (.mkv)
- **video/quicktime** (.mov)
- **video/mp4** (.mp4)
- **video/mpeg** (.mpeg)
- **video/webm** (.webm)
- **video/x-ms-wmv** (.wmv)
- **audio/mp3** (.mp3)
- **audio/ogg** (.ogg)
- **audio/flac** (.flac)
- **audio/x-ms-wma** (.wma)
- **application/x-shockwave-flash** (.swf)
- **application/pdf** (.pdf)
- **application/x-photoshop** (.psd)
- **application/clip** (.clip)
- **application/vnd.rar** (.rar)
- **application/zip** (.zip)
- **application/x-7z-compressed** (.7z)

Although some support is imperfect for the complicated filetypes. For the Windows and Linux built releases, hydrus now embeds an MPV player for video, audio and gifs, which provides smooth playback and audio, but some other environments may not support MPV and so will default when possible to the native hydrus software renderer, which does not support audio. When something does not render how you want, right-clicking on its thumbnail presents the option 'open externally', which will open the file in the appropriate default program (e.g. ACDSee, VLC).

The client can also download files from several websites, including 4chan and other imageboards, many boorus, and gallery sites like deviant art and hentai foundry. You will learn more about this later.

inbox and archiving

The client sends newly imported files to an **inbox**, just like your email. Inbox acts like a tag, matched by 'system:inbox'. A small envelope icon is drawn in the top corner of all inbox files:



If you are sure you want to keep a file long-term, you should **archive** it, which will remove it from the inbox. You can archive from your selected thumbnails' right-click menu, or by pressing F7. If you make a mistake, you can spam Ctrl-Z for undo or hit Shift-F7 on any set of files to explicitly return them to the inbox.

Anything you do not want to keep should be deleted by selecting from the right-click menu or by hitting the delete key. Deleted files are sent to the trash. They will get a little trash icon:



A trashed file will not appear in subsequent normal searches, although you can search the trash specifically by clicking the 'my files' button on the autocomplete dropdown and changing the file domain to 'trash'. Undeleting a file (shift+delete) will return it to 'my files' as if nothing had happened. Files that remain in the trash will be permanently deleted, usually after a few days. You can change the permanent deletion behaviour in the client's options.

A quick way of processing new files is--

filtering

Lets say you just downloaded a good thread, or perhaps you just imported an old folder of miscellany. You now have a whole bunch of files in your inbox--some good, some awful. You probably want to quickly go through them, saying *yes, yes, yes, no, yes, no, no, yes*, where *yes* means 'keep and archive' and *no* means 'delete this trash'. **Filtering** is the solution.

Select some thumbnails, and either choose *filter->archive/delete* from the right-click menu or hit F12. You will see them in a special version of the media viewer, with the following controls:

- Left-click, space, or F7: **keep and archive the file, move on**
- Right-click or delete: **delete the file, move on**
- Arrow key up: **Skip this file, move on**
- Middle-click or backspace: **I didn't mean that, go back one**
- Escape, return, or F12: **stop filtering now**

Your choices will not be committed until you finish filtering.

This saves time.

lastly

The hydrus client's workflows are not designed for half-finished files that you are still working on. Think of it as a giant archive for everything excellent you have decided to store away. It lets you find and remember these things quickly.

In general, hydrus is good for individual files like you commonly find on imageboards or boorus. Although advanced users can cobble together some page-tag-based solutions, it is not yet great for multi-file media like comics and definitely not as a typical playlist-based music player.

If you are looking for a comic manager to supplement hydrus, check out this user-made guide to other archiving software [here](#)!

And although the client can hold millions of files, it starts to creak and chug when displaying or otherwise tracking more than about 40,000 or so in a single gui window. As you learn to use it, please try not to let your download queues or general search pages regularly sit at more than 40 or 50k total *items*, or you'll start to slow other things down. Another common mistake is to leave one large 'system:everything' or 'system:inbox' page open with 70k+ files. For these sorts of 'ongoing processing' pages, try adding a 'system:limit=256' to keep them snappy. One user mentioned he had regular gui hangs of thirty seconds or so, and when we looked into it, it turned out his handful of download pages had three million files queued up! Just try and take things slow until you figure out what your computer's limits are.

Getting started: Tags

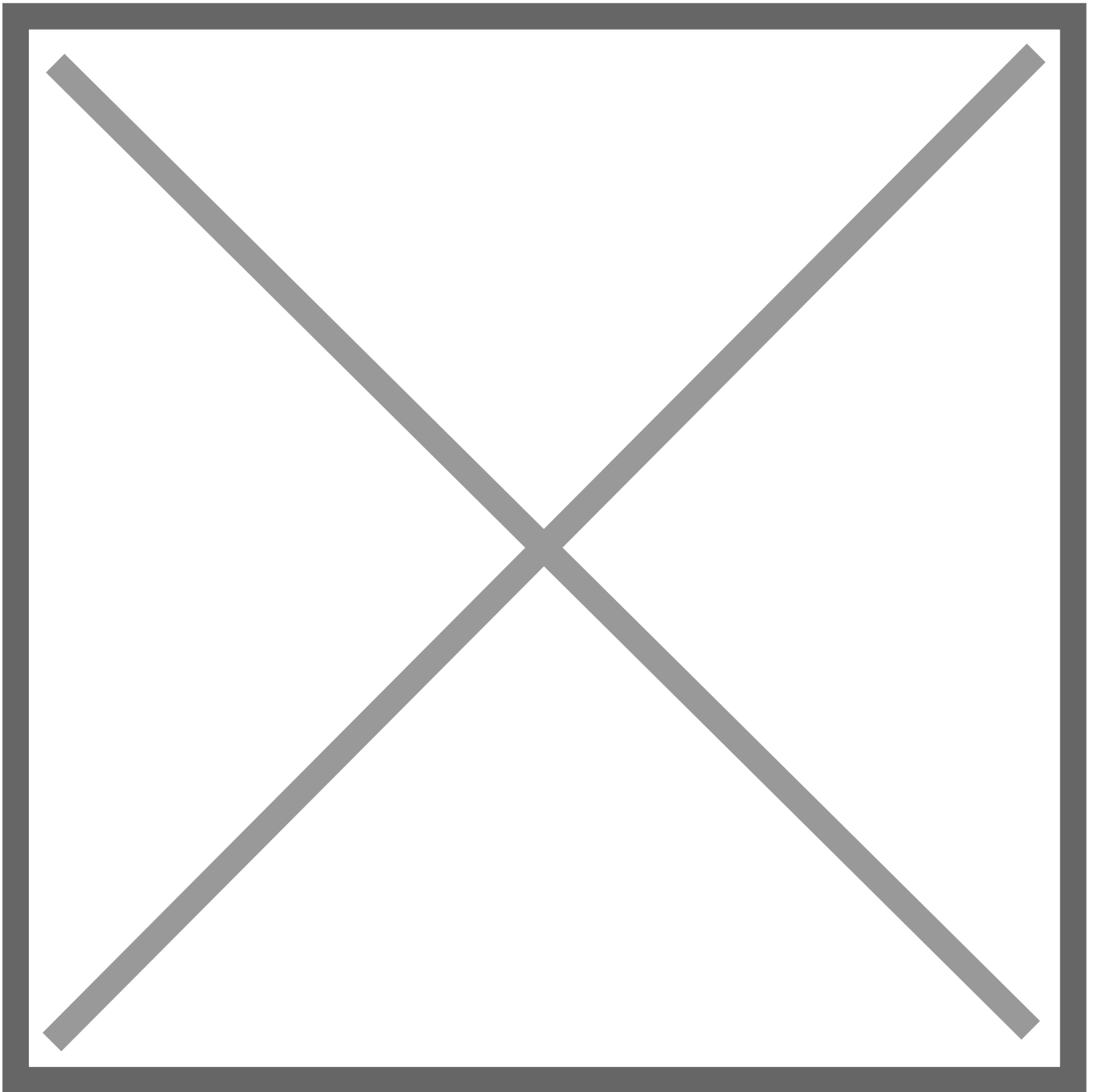
If any of this is confusing, a simpler guide is [here](#), and some video guides are [here](#)!

how do we find files?

So, you have stored some media in your database. Everything is hashed and cached. You can search by inbox and resolution and size and so on, but if you really want to find what we are looking for, you will have to use *tags*.

FAQ: what is a tag?

Your client starts with one local tags service, called 'my tags', which keeps all of its file->tag mappings in your client's database where only you can see them. It is a good place to practise. So, select a file and press F3:



The autocomplete dropdown in the manage tags dialog works very like the one in a normal search page--you type part of a tag, and matching results will appear below. You select the tag you want with the arrow keys and hit enter. Since your 'my tags' service doesn't have any tags in it yet, you won't get any results here except the exact match of what you typed. If you want to remove a tag, enter the exact same thing again or double-click it in the box above.

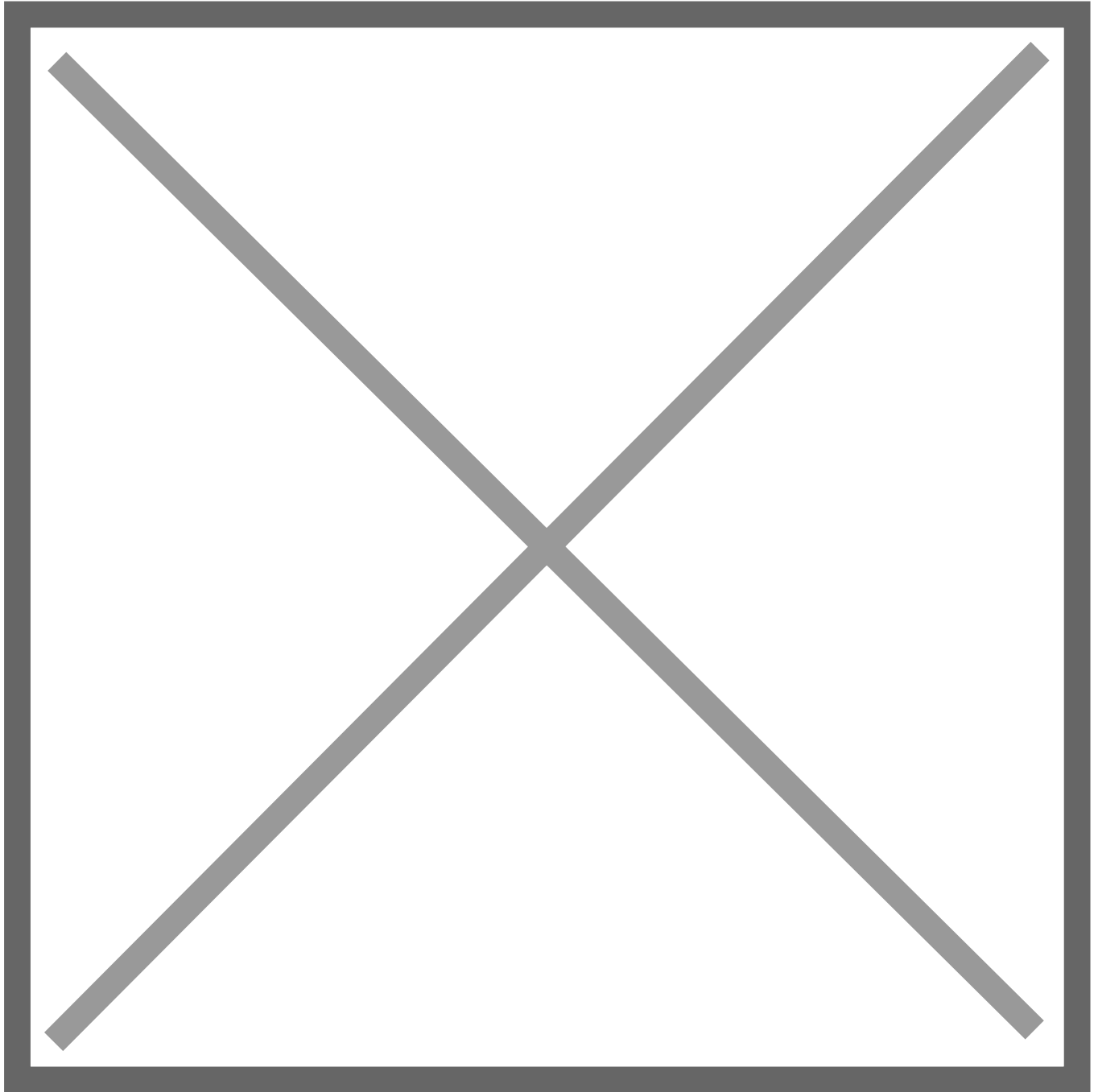
Prefixing a tag with a category and a colon will create a *namespaced* tag. This helps inform the software and other users about what the tag is. Examples of namespaced tags are:

- character:batman
- series:street fighter
- person:jennifer lawrence

- title:vitruvian man

The client is set up to draw common namespaces in different colours, just like boorus do. You can change these colours in the options.

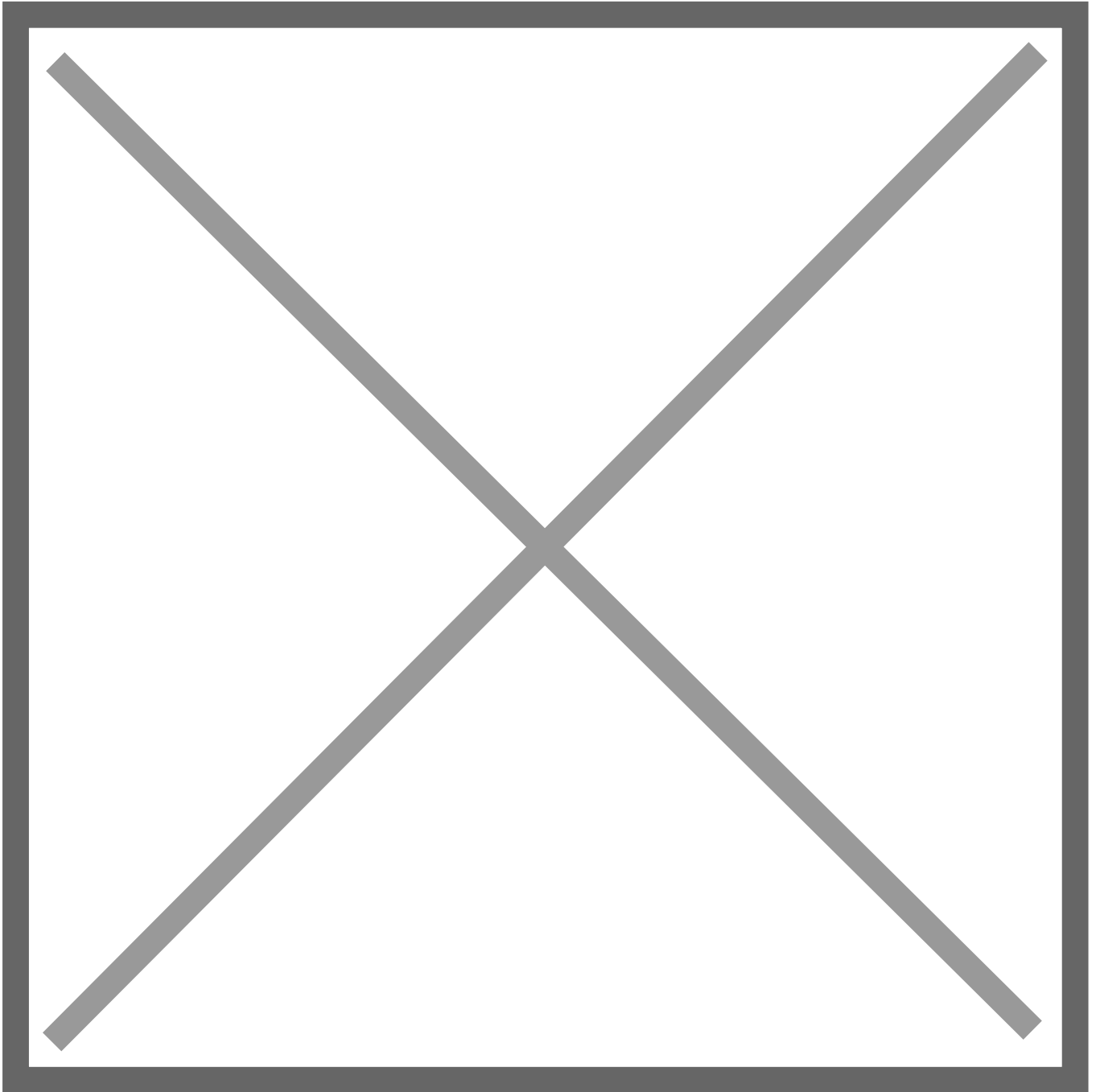
Once you are happy with your tags, hit 'apply' or just press enter on the text box if it is empty.



The tags are now saved to your database. Searching for any of them will return this file and anything else so tagged:



If you add more tags or system predicates to a search, you will limit the results to those files that match every single one:



You can also exclude a tag by prefixing it with a hyphen (e.g. '-heresy').

OR searching

Searches find files that match every search 'predicate' in the list (it is an **AND** search), which makes it difficult to search for files that include one **OR** another tag. More recently, simple OR search support was added. All you have to do is hold down Shift when you enter/double-click a tag in the autocomplete entry area. Instead of sending the tag up to the active search list up top, it will instead start an under-construction 'OR chain' in the tag results below:



You can keep searching for and entering new tags. Holding down Shift on new tags will extend the OR chain, and entering them as normal will 'cap' the chain and send it to the complete and active search predicates above.



Any file that has one or more of those OR sub-tags will match.

If you enter an OR tag incorrectly, you can either cancel or 'rewind' the under-construction search predicate with these new buttons that will appear:



You can also cancel an under-construction OR by hitting Esc on an empty input. You can add any sort of search term to an OR search predicate, including system predicates. Some unusual sub-predicates (typically a '-tag', or a very broad system predicate) can run very slowly, but they will run much faster if you include non-OR search predicates in the search:



This search will return all files that have the tag 'fanfic' and one or more of 'medium:text', a positive value for the like/dislike rating 'read later', or PDF mime.

tag repositories

It can take a long time to tag even small numbers of files well, so I created *tag repositories* so people can share the work.

Tag repos store many file->tag relationships. Anyone who has an access key to the repository can sync with it and hence download all these relationships. If any of their own files match up, they will get those tags. Access keys will also usually have permission to upload new tags and ask for incorrect existing ones to be deleted.

Anyone can run a tag repository, but it is a bit complicated for new users. I ran a public tag repository for a long time, and now this large central store is run by users. It has hundreds of millions of tags and is free to access and contribute to.

To connect with it, please check [here](#).

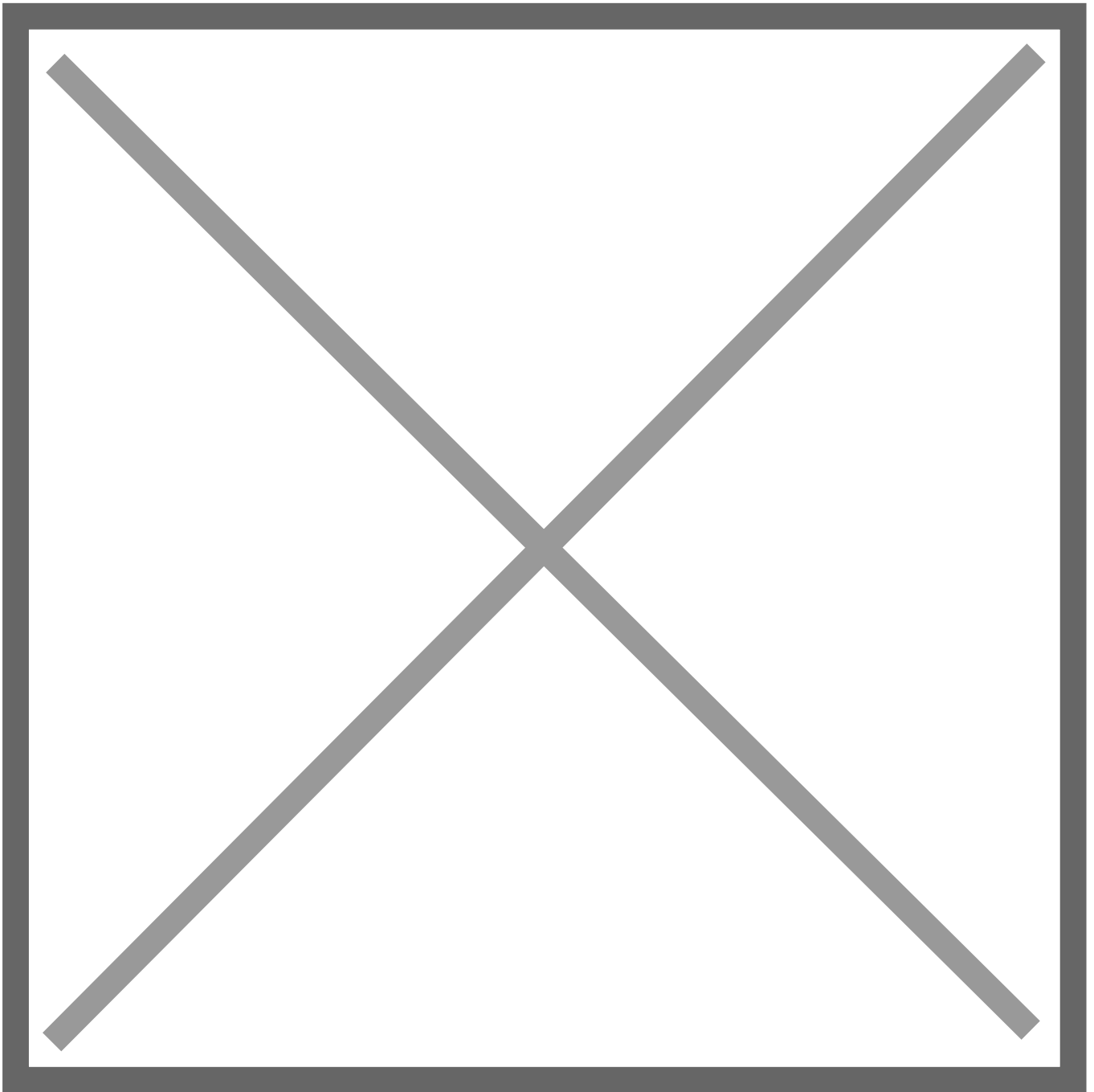
If you add it, your client will download updates from the repository over time and, usually when it is idle or shutting down, 'process' them into its database until it is fully synchronised. The processing step is CPU and HDD heavy, and you can customise when it happens in *file->options->maintenance and processing*. As the repository synchronises, you should see some new tags appear, particularly on famous files that lots of people have.

Tags are rich, cpu-intensive metadata. The Public Tag Repository has hundreds of millions of mappings, and your client will eventually download and index them all. Be aware that the PTR has been growing since 2011 and now has hundreds of millions of mappings. As of 2020-03, it requires about 4GB of bandwidth and file storage, and your database itself will grow by 25GB! It will take *hours* of total processing time to fully synchronise. Because of mechanical drive latency, HDDs are often too slow to process hundreds of millions of tags in reasonable time. Syncing with large repositories is only recommended if your hydrus db is on an SSD. Even then, it is best left to work on this in small pieces in the background, either during idle time or shutdown time, so unless you are an advanced user, just leave it to download and process on its own--it usually takes a couple of weeks to quietly catch up.

You can watch more detailed synchronisation progress in the *services->review services* window.



Your new service should now be listed on the left of the manage tags dialog. Adding tags to a repository works very similarly to the 'my tags' service except hitting 'apply' will not immediately confirm your changes--it will put them in a queue to be uploaded. These 'pending' tags will be counted with a plus '+' or minus '-' sign:



Notice that a 'pending' menu has appeared on the main window. This lets you start the upload when you are ready and happy with everything that you have queued.

When you upload your pending tags, they will commit and look to you like any other tag. The tag repository will anonymously bundle them into the next update, which everyone else will download in a day or so. They will see your tags just like you saw theirs.

If you attempt to remove a tag that has been uploaded, you may be prompted to give a reason, creating a petition that a janitor for the repository will review.

I recommend you not spam tags to the public tag repo until you get a rough feel for the [guidelines](#), and my original [tag schema](#) thoughts, or just lurk until you get the idea. It roughly follows what you

will see on a typical booru. The general rule is to only add factual tags--no subjective opinion.

You can connect to more than one tag repository if you like. When you are in the *manage tags* dialog, pressing the up or down arrow keys on an empty input switches between your services.

FAQ: why can my friend not see what I just uploaded?

Getting started: Downloading

downloading

The hydrus client has a sophisticated and completely user-customisable download system. It can pull from any booru or regular gallery site or imageboard, and also from some special examples like twitter and tumblr. A fresh install will by default have support for the bigger sites, but it *is* possible, with some work, for any user to create a new shareable downloader for a new site.

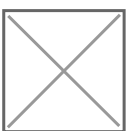
The downloader is highly parallelisable, and while the default bandwidth rules should stop you from running too hot and downloading so much at once that you annoy the servers you are downloading from, there are no brakes in the program on what you can get.

It is very important that you take this slow. Many users get overexcited with their new ability to download 500,000 files *and then do so*, only discovering later that 98% of what they got was junk that they now have to wade through. Figure out what workflows work for you, how fast you process files, what content you *actually* want, how much bandwidth and hard drive space you have, and prioritise and throttle your incoming downloads to match. If you can realistically only archive/delete filter 50 files a day, there is little benefit to downloading 500 new files a day. START SLOW.

It also takes a decent whack of CPU to import a file. You'll usually never notice this with just one hard drive import going, but if you have twenty different download queues all competing for database access and individual 0.1-second hits of heavy CPU work, you will discover your client starts to judder and lag. Keep it in mind, and you'll figure out what your computer is happy with. I also recommend you try to keep your total loaded files/urls to be under 20,000 to keep things snappy. Remember that you can pause your import queues, if you need to calm things down a bit.

let's do it

Open the new page selector with F9 and then hit *download->gallery*:




You can do a test download here of a few files if you want, but don't start downloading loads of stuff until you have read about parsing tags!

The gallery page can download from multiple sources at the same time. Each entry in the list represents a basic combination of two things:

- **source** - The site you are getting from. Safebooru or Danbooru or Deviant Art or twitter or anywhere else.
- **query text** - Something like 'contrapposto' or 'blonde_hair blue_eyes' or an artist name like 'incase'. Whatever is searched on the site to return a list of ordered media.

So, when you want to start a new download, you first select the source with the button--by default, it is probably 'Artstation' for you--and then type in a query in the text box and hit enter. The download will soon start and fill in information, and thumbnails should stream in, just like the hard drive importer. The downloader typically works by walking through the search's gallery pages one by one, queueing up the found files for later download. There are several intentional delays built into the system, so do not worry if work seems to halt for a little while--you will get a feel for it with experience.

The thumbnail panel can only show results from one queue at a time, so double-click on an entry to 'highlight' it, which will show its thumbs and also give more detailed info and controls in the 'highlighted query' panel. I encourage you to explore the highlight panel over time, as it can show and do quite a lot. Double-click again to 'clear' it.

It is a good idea to 'test' larger downloads, either by visiting the site itself for that query, or just waiting a bit and reviewing the first files that come in. Just make sure that you *are* getting what you thought you would, whether that be verifying that the query text is correct or that the site isn't only giving you bloated gifs or other bad quality files. The 'file limit', which stops the gallery search after the set number of files, is also great for limiting fishing expeditions (such as overbroad searches like 'wide_hips', which on the bigger boorus have 100k+ results and return *variable* quality). If the gallery search runs out of new files before the file limit is hit, the search will naturally stop (and the entry in the list should gain a  'stop' symbol).

Note that some sites only serve 25 or 50 pages of results, despite their indices suggesting hundreds. If you notice that one site always bombs out at, say, 500 results, it may be due to a decision on their end. You can usually test this by visiting the pages hydrus tried in your web browser.

In general, particularly when starting out, artist searches are best. They are usually fewer than a thousand files and have fairly uniform quality throughout.

parsing tags

But we don't just want files--most sites offer tags as well. **By default, hydrus does not fetch any tags for downloads.** As you use the client, you will figure out what sorts of tags you are

interested in and shape your parsing rules appropriately, but for now, let's do a test that just gets everything--click *tag import options*:



By default, all 'tag import options' objects defer to the client's defaults. Since we want to change this queue from the current default of 'get nothing' to 'get everything', uncheck the top default checkbox and then click 'get tags' on a tag service, whether that is your 'my tags' or the PTR if you have added it. Hit apply and run a simple query for something, like 'blue_eyes' on one of the boorus. Pause its gallery search after a page or two, and then pause the import queue after a dozen or so files come in--they should be really well tagged!

It is easy to get tens of thousands of tags this way. Different sites offer different kinds and qualities of tags, and the client's downloaders (which were designed by me, the dev, or a user) may parse all or only some of them. Many users like to just get everything on offer, but others only ever want, say, 'creator', 'series', and 'character' tags. If you feel brave, click that 'all tags' button on tag import options, which will take you into hydrus's advanced 'tag filter', which allows you to whitelist or blacklist the incoming list of tags according to whatever your preferences are.

The file limit and file/tag import options on the upper panel, if changed, will only apply to **new** queries. If you want to change the options for an existing queue, either do so on its highlight panel or use the 'set options to queries' button.

Tag import options can get complicated. The blacklist button will let you skip downloading files that have certain tags (perhaps you would like to auto-skip all images with 'gore', 'scat', or 'diaper?'), again using the tag filter. The 'additional tags' also allow you to add some personal tags to all files coming in--for instance, you might like to add 'process into favourites' to your 'my tags' for some query you really like so you can find those files again later and process them separately. That little 'cog' icon button can also do some advanced things. I recommend you start by just getting everything (or nothing, if you really would rather tag everything yourself), and then revisiting it once you have some more experience. Once you have played with this a bit, let's fix your preferences as the new default:

default tag import options

Hit *network->downloaders->manage default tag import options*. Set a new default for 'file posts', and that will be the default (that we originally turned off above) for all gallery download pages (and subscriptions, which you will learn about later). You can have different TIOs for each site, but again, we will leave it simple for now.

watching threads

If you are an imageboard user, try going to a thread you like and drag-and-drop its URL (straight from your web browser's address bar) onto the hydrus client. It should open up a new 'watcher' page and import the thread's files!



With only one URL to check, watchers are a little simpler than gallery searches, but as that page is likely receiving frequent updates, it checks it over and over until it dies. By default, the watcher's 'checker options' will regulate how quickly it checks based on the speed at which new files are coming in--if a thread is fast, it will check frequently; if it is running slow, it may only check once per day. When a thread falls below a critical posting velocity or 404s, checking stops.

In general, you can leave the checker options alone, but you might like to revisit them if you are always visiting faster or slower boards and find you are missing files or getting DEAD too early.

bandwidth

It will not be too long until you see a "bandwidth free in xxxxx..." message. As a long-term storage solution, hydrus is designed to be polite in its downloading--both to the source server and your computer. The client's default bandwidth rules have some caps to stop big mistakes, spread out larger jobs, and at a bare minimum, no domain will be hit more than once a second.

All the bandwidth rules are completely customisable. They can get quite complicated. I **strongly** recommend you not look for them until you have more experience. I **especially strongly** recommend you not ever turn them all off, thinking that will improve something, as you'll probably render the client too laggy to function and get yourself an IP ban from the next server you pull from.

If you want to download 10,000 files, set up the queue and let it work. The client will take breaks, likely even to the next day, but it will get there in time. Many users like to leave their clients on all the time, just running in the background, which makes these sorts of downloads a breeze--you check back in the evening and discover your download queues, watchers, and subscriptions have given you another thousand things to deal with.

Again: the real problem with downloading is not finding new things, it is keeping up with what you get. Start slow and figure out what is important to your bandwidth budget, hard drive budget, and free time budget. Almost everyone fails at this.

subscriptions

Subscriptions are a way to automatically recheck a good query in future, to keep up with new files. Many users come to use them. When you are comfortable with downloaders and have an idea of

what you like, come back and read the subscription help, which is [here](#).

other downloading

There are two other ways of downloading, mostly for advanced or one-off use.

The **url downloader** works like the gallery downloader but does not do searches. You can paste downloadable URLs to it, and it will work through them as one list. Dragging and dropping recognisable URLs onto the client (e.g. from your web browser) will also spawn and use this downloader.

The **simple downloader** will do very simple parsing for unusual jobs. If you want to download all the images in a page, or all the image link destinations, this is the one to use. There are several default parsing rules to choose from, and if you learn the downloader system yourself, it will be easy to make more.

logins

The client now supports a flexible (but slightly prototype and ugly) login system. It can handle simple sites and is as completely user-customisable as the downloader system. The client starts with multiple login scripts by default, which you can review under *network->downloaders->manage logins*:



Many sites grant all their content without you having to log in at all, but others require it for NSFW or special content, or you may wish to take advantage of site-side user preferences like personal blacklists. If you wish, you can give hydrus some login details here, and it will try to login--just as a browser would--before it downloads anything from that domain.

For multiple reasons, I do not recommend you use important accounts with hydrus. Use a throwaway account you don't care much about.

To start using a login script, select the domain and click 'edit credentials'. You'll put in your username/password, and then 'activate' the login for the domain, and that should be it! The next time you try to get something from that site, the first request will wait (usually about ten seconds) while a login popup performs the login. Most logins last for about thirty days (and many refresh that 30-day timer every time you make a new request), so once you are set up, you usually never notice it again, especially if you have a subscription on the domain.

Most sites only have one way of logging in, but hydrus does support more. Hentai Foundry is a good example--by default, the client performs the 'click-through' login as a guest, which requires no credentials and means any hydrus client can get any content from the start. But this way of logging in only lasts about 60 minutes or so before having to be refreshed, and it does not hide any spicy stuff, so if you use HF a lot, I recommend you create a throwaway account, set the filters you like in your HF profile (e.g. no guro content), and then click the 'change login script' in the client to the proper username/pass login.

The login system is new and still a bit experimental. Don't try to pull off anything too weird with it! If anything goes wrong, it will likely delay the script (and hence the whole domain) from working for a while, or invalidate it entirely. If the error is something simple, like a password typo or current server maintenance, go back to this dialog to fix and scrub the error and try again. If the site just changed its layout, you may need to update the login script. If it is more complicated, please contact me, hydrus_dev, with the details!

If you would like to login to a site that is not yet supported by hydrus (usually ones with a Captcha in the login page), see about getting a web browser add-on that lets you export a cookies.txt (either for the whole browser or just for that domain) and then drag and drop that file onto the hydrus *network->data->review session cookies* dialog. This sometimes does not work if your add-on's export formatting is unusual. If it does work, hydrus will import and use those cookies, which skips the login by making your hydrus pretend to be your browser directly. This is obviously advanced and hacky, so if you need to do it, let me know how you get on and what tools you find work best!

Getting started: Ratings

The hydrus client supports two kinds of ratings: *like/dislike* and *numerical*. Let's start with the simpler one:

like/dislike

This can set one of two values to a file. It does not have to represent like or dislike--it can be anything you want. Go to *services->manage services->local->like/dislike ratings*:



You can set a variety of colours and shapes.

numerical

This is '3 out of 5 stars' or '8/10'. You can set the range to whatever whole numbers you like:



As well as the shape and colour options, you can set how many 'stars' to display and whether 0/10 is permitted.

If you change the star range at a later date, any existing ratings will be 'stretched' across the new range. As values are collapsed to the nearest integer, this is best done for scales that are multiples. 2/5 will neatly become 4/10 on a zero-allowed service, for instance, and 0/4 can nicely become 1/5 if you disallow zero ratings in the same step. If you didn't intuitively understand that, just don't touch the number of stars or zero rating checkbox after you have created the numerical rating service!

now what?

Ratings are displayed in the top-right of the media viewer:



Hovering over each control will pop up its name, in case you forget which is which. You can set then them with a left- or right-click. Like/dislike and numerical have slightly different click behaviour, so have a play with them to get their feel. Pressing F4 on a selection of thumbnails will open a dialog with a very similar layout, which will let you set the same rating to many files simultaneously.

Once you have some ratings set, you can search for them using `system:rating`, which produces this dialog:



On my own client, I find it useful to have several like/dislike ratings set up as one-click pseudo-tags, like the 'OP images' above.

Access keys

The PTR is now run by users with more bandwidth than I had to give, so the bandwidth limits are gone! If you would like to talk with the new management, please check the [discord](#).

A guide and schema for the new PTR is [here](#).

first off

I have purposely not pre-baked any default repositories into the client. You have to choose to connect yourself. **The client will never connect anywhere until you tell it to.**

For a long time, I ran the Public Tag Repository. It grew to 650 million tags and I no longer had the bandwidth or janitor time it deserved. It is now run by users.

I created a 'frozen' copy of the PTR when I stopped running it. If you are an advanced user, you can run your own new tag repository starting from that frozen point or, if you know python or SQLite and wish to play around with its data, get more easily accessible Hydrus Tag Archives of its tags and siblings and pairs, right [here](#).

easy setup

Hit *help->add the public tag repository* and you will all be set up.

manually

To add a new repository to your client, hit *services->manage services* and click the add button:



Here's the info so you can copy it:

- ptr.hydrus.network
- 45871
- 4a285629721ca442541ef2c15ea17d1f7f7578b0c3f4f5f2a05f8f0ab297786f

It is worth checking the 'test address' and 'test access key' buttons just to double-check your firewall and key are all correct.

Tags are rich, cpu-intensive metadata. The Public Tag Repository has hundreds of millions of mappings, and your client will eventually download and index them all. Be aware that the PTR has been growing since 2011 and now has hundreds of millions of mappings. As of 2020-03, it requires about 4GB of bandwidth and file storage, and your database itself will grow by 25GB! It will take *hours* of total processing time to fully synchronise. Because of mechanical drive latency, HDDs are often too slow to process hundreds of millions of tags in reasonable time. Syncing with large repositories is only recommended if your hydrus db is on an SSD. Even then, it is best left to work on this in small pieces in the background, either during idle time or shutdown time, so unless you are an advanced user, just leave it to download and process on its own--it usually takes a couple of weeks to quietly catch up.

jump-starting an install

A user kindly manages a store of update files and pre-processed empty client databases to get your synced quicker. This is generally recommended for advanced users or those following a guide, but if you are otherwise interested, please check it out:

<https://cuddlebear92.github.io/Quicksync/>