The Next Step

- more getting started with files
- adding new downloaders
- thoughts on a public tagging schema
- Getting started with subscriptions
- Filtering Duplicates
- Reducing program lag

more getting started with files

exporting and uploading

There are many ways to export files from the client:

- **drag and drop**
  Just dragging from the thumbnail view will export (copy) all the selected files to wherever you drop them.

  The files will be named by their ugly hexadecimal hash, which is how they are stored inside the database.
  If you use this to open a file inside an image editing program, remember to go 'save as' and give it a new filename! The client does not expect files inside its db directory to change.
- **export dialog**
  Right clicking some files and selecting *share->export->files* will open this dialog:
  
  Which lets you export the selected files with custom filenames. It will initialise trying to export the files named by their hashes, but once you are comfortable with tags, you'll be able to generate much cleverer and prettier filenames.
- **share->copy->files**
  This will copy the files themselves to your clipboard. You can then paste them wherever you like, just as with normal files. They will have their hashes for filenames.
  This is a very quick operation. It can also be triggered by hitting Ctrl+C.
- **share->copy->hashes**
  This will copy the files' unique identifiers to your clipboard, in hexadecimal.
  You will not have to do this often. It is best when you want to identify a number of files to someone else without having to send them the actual files.

adding new downloaders

all downloaders are user-creatable and - shareable

Since the big downloader overhaul, all downloaders can be created, edited, and shared by any user. Creating one from scratch is not simple, and it takes a little technical knowledge, but importing what someone else has created is easy.

Hydrus objects like downloaders can sometimes be shared as data encoded into png files, like this:

This contains all the information needed for a client to add a realbooru tag search entry to the list you select from when you start a new download or subscription.

You can get these pngs from anyone who has experience in the downloader system. An archive is maintained here.

To 'add' the easy-import pngs to your client, hit *network->downloaders->import downloaders*. A little image-panel will appear onto which you can drag-and-drop these png files. The client will then decode and go through the png, looking for interesting new objects and automatically import and link them up without you having to do any more. Your only further input on your end is a 'does this look correct?' check right before the actual import, just to make sure there isn't some mistake or other glaring problem.

Objects imported this way will take precedence over existing functionality, so if one of your downloaders breaks due to a site change, importing a fixed png here will overwrite the broken entries and become the new default.

thoughts on a public tagging schema

This document was originally written for when I ran the Public Tag Repository. This is now run by users, so I am no longer an authority for it. I am briefly editing the page and leaving it as a record for some of my thoughts on tagging if you are interested. You can, of course, run your own tag repositories and do your own thing additionally or instead.

A newer guide and schema for the PTR is here.

seriousness of schema

This is not all that important; it just makes searches and cooperation easier if most of us can mostly follow some guidelines.

We will never be able to easily and perfectly categorise every single image to everyone's satisfaction, so there is no point defining every possible rule for every possible situation. If you do something that doesn't fit, fixing mistakes is not difficult.

If you are still not confident, just lurk for a bit. See how other people have tagged the popular images and do more of that.

you can add pretty much whatever the hell you want, but don't screw around

The most important thing is: **if your tag is your opinion, don't add it**. 'beautiful' is an unhelpful tag because no one can agree on what it means. 'lingerie', 'blue eyes', and 'male' or 'female' are better since reasonable people can generally agree on what they mean. If someone thinks blue-eyed women are beautiful, they can search for that to find beautiful things.

You can start your own namespaces, categorisation systems, whatever. Just be aware that everyone else will see what you do.

If you are still unsure about the difference between objective and subjective, here's some more examples:

- objective tags: (add these!)
  - firetruck
  - hors d'œuvre
  - high heels
  - character:jean-luc picard
  - person:okita anri
  - title:the tragical history of hamlet, prince of denmark
  - page:17
- subjective tags: (don't add these!)
  - awesome
  - faggot level:super-gay
  -
  - 4 stars
  - this is boring, why did anyone upload this here
  - moran communist and ONE TERM PRESIDENT!!! SARAH PALIN 2012! FOR JESUS CHRIST

Of course, if you are tagging a picture of someone holding a sign that says 'beautiful', you can bend the rules. Otherwise, please keep your opinions to yourself!

numbers

Numbers should be written '22', '1457 ce', and 'page:3', unless as part of an official title like 'ocean's eleven'. When the client parses and sorts numbers, it does so intelligently, so just use '1' where you might before have done '01' or '001'. I know it looks ugly sometimes to have '2 girls' or '1 cup', but the rules for writing numbers out in full are hazy for special cases.

(Numbers written as 123 are also readable by many different language-speakers, while 'tano', 'deux' and 'seven' are not.)

plurals

Nouns should generally be singular, not plural. 'chair' instead of 'chairs', 'cat' instead of 'cats', even if there are several of the thing in the image. If there really are *many* of the thing in the image, add a seperate 'multiple' or 'lineup' tag as appppropriate.

Ignore this when the thing is normally said in its plural (usually paired) form. Say 'blue eyes', not 'blue eye'; 'breasts', not 'breast', even if only one is pictured.

acronyms and synonyms

I personally prefer the full 'series:the lord of the rings' rather than 'lotr'. If you are an advanced user, please help out with tag siblings to help induce this.

character:anna (frozen)

I am not fond of putting a series name after a character because it looks unusual and is applied unreliably. It is done to separate same-named characters from each other (particularly when they have no canon surname), which is useful in places that search slowly, have thin tag areas on their web pages, or usually only deal in single-tag searches. For archival purposes, I generally prefer that namespaces are stored as the namespace and nowhere else. 'series:harry potter' and 'character:harry potter', not 'harry potter (harry potter)'. Some sites even say things like 'anna (disney)'. It isn't a big deal, but if you are adding a sibling to collapse these divergent tags into the 'proper' one, I'd prefer it all went to the simple and reliable 'character:anna'. Even better would be migrating towards a canon-ok unique name, like 'character:princess anna of arendelle', which could have the parent 'series:frozen'.

Including nicknames, like 'character:angela "mercy" ziegler' can be useful to establish uniqueness, but are not mandatory. 'character:harleen "harley quinn" frances quinzel' is probably overboard.

protip: rein in your spergitude

In developing hydrus, I have discovered two rules to happy tagging:

1. Don't try to be perfect.
2. Only add those tags you actually use in searches.

Tagging can be fun, but it can also be complicated, and the problem space is gigantic. There is always works to do, and it is easy to exhaust onesself or get lost in the bushes agonising over whether to use 'smile' or 'smiling' or 'smirk' or one of a million other split hairs. Problems are easy to fix, and this marathon will never finish, so do not try to sprint. The ride never ends.

The sheer number of tags can also be overwhelming. Importing all the many tags from the boorus is totally fine, but if you are typing tags yourself, I suggest you try not to exhaustively tag everything in the image. You will save a lot of time and ultimately be much happier with your work. Anyone can see what is in an image just by looking at it--tags are primarily for finding things. Character, series and creator namespaces are a great place to start. After that, add what you are interested in, be that 'blue sky' or 'midriff'.

newer thoughts on presentation preferences

Since developing and receiving feedback for the siblings system, and then in dealing with siblings with the PTR, I have come to believe that the most difficult disagreement to resolve in tagging is not in what is in an image, but how those tags should present. It is easy to agree that an image contains a 'bikini', but should that show as 'bikini' or 'clothing:bikini' or 'general:bikini' or 'swimwear:bikini'? Which is better?

This is impossible to answer definitively. There is no perfect dictionary that satisfies everyone, and opinions are fairly fixed. My intentions for future versions of the sibling and tag systems is to allow users to broadly tell the client some display rules such as 'Whenever you have a clothing: tag, display it as unnamespaced' and eventually more sophisticated ones like 'I prefer slang, so show pussy instead of vagina'.

siblings and parents

Please do add siblings and parents! If it is something not obvious, please explain the relationship in your submitted reason. If it *is* something obvious (e.g. 'wings' is a parent of 'angel wings'), don't bother to put a reason in; I'll just approve it.

My general thoughts:

- ## siblings

  In general, the correctness of a thing is in how it would describe itself, or how its creator would describe it.
  For shorthand, I will say 'a'->'b' to mean 'a' is replaced by 'b'.
  For instance, japanese names are usually written surname first and western forename first, so let's go 'character:rei ayanami'->'character:ayanami rei' but leave 'person:emma watson' and other western names as they are.
  Unless it is too obscure, replace the english version of a word with any more proper or original foreign name. But stick to something a westerner can read. Do things like 'series:the melancholy of haruhi suzumiya'->'series:haruhi suzumiya no yuuutsu' or 'series:princess mononoke'->'series:mononoke hime'. There's even an argument for things like 'series:harry potter and the sorcerer's stone'->'series:harry potter and the philosopher's stone'.

Accents and other unusual/unicode characters are great in tags if they reflect the official marketed name, and should be preferred, but make sure there's an ascii->unicode sibling to make it easy for most users to type. 'series:pokemon'->'series:pok&#65533;mon' is excellent, as it both reflects official branding and also helps anyone who can't easily produce '&#65533;' on their keyboard find it.
I don't care about popularity as much as accuracy. Given 'series:pretty cure' and 'series:precure', I would prefer 'series:pretty cure' because it is the 'full and proper' rendering, even though there are more instances of 'precure' on the boorus.
Do correct for common plural mistakes. ear->ears, women->female, and so on.
And feel free to replace any 'character (series)' booru artifacts as with the 'anna (frozen)' example above. 'character:anna (frozen)'->'character:princess anna of arendelle' is great wherever it makes sense.
But please **do not** go 'blah'->'character:blah' unless the name is popular and unique. No one is going to be confused by 'ayanami rei'->'character:ayanami rei', but going 'archer'->'character:archer' is going to create a lot of false positives. There's a similar problem with something like 'character:mercy'->'character:angela "mercy" ziegler'--although the left hand side is namespaced, there are still plenty of *characters* named 'mercy', so a sibling that converts all Mercys to Overwatch's Mercy is not appropriate.
If the character name is the same as the series name, make the unnamespaced version go to the series version. For instance, set 'harry potter'->'series:harry potter', since we don't know which one it is and 'character:harry potter' ⊂ 'series:harry potter'. (If a picture of just Hermione that for some reason was not providing namespace information had 'hermione granger' (the character) and 'harry potter' (the series), we wouldn't want to infer 'character:harry potter' by accident.
In general, swap out slang for proper terms. 'lube'->'lubricant', 'series:zelda'->'series:the legend of zelda'.

- # parents

Be shy about adding character:blah->series:whatever unless you are certain the character name is unique. 'character:harry potter'->'series:harry potter' seems fairly uncontroversial, for instance, but adding specific sub-series just to be completionist, such as 'character:miranda lawson->series:mass effect: redemption' is asking for trouble.
Remember that parents define a relationship that is always true. Don't add 'blonde hair' to 'character:elsa', even though it is true in most files--add 'animal ears' to 'cat ears', as cat ears are always animal ears, no matter what an artist can think up.
Also, tag parents are only worth something if the parent is useful for searching. Adding 'medium:blue background'->'blue' isn't useful since 'blue' itself is not very valuable, but 'fishnet stockings'->'stockings' is useful as both tags are common and used in searches by plenty of people.
You can create a complicated tree like the firearms diagram on my parents page, but if it only adds seven tags that you probably wouldn't ever use yourself, you probably wasted your time.

Getting started with subscriptions

Do not try to create a subscription until you are comfortable with a normal gallery download page! Go here.

Let's say you found an artist you like. You downloaded everything of theirs from some site, but one or two pieces of new work is posted every week. You'd like to keep up with the new stuff, but you don't want to manually make a new download job every week for every single artist you like.

what are subs?

Subscriptions are a way of telling the client to regularly and quietly repeat a gallery search. You set up a number of saved queries, and the client will 'sync' with the latest files in the gallery and download anything new, just as if you were running the download yourself.

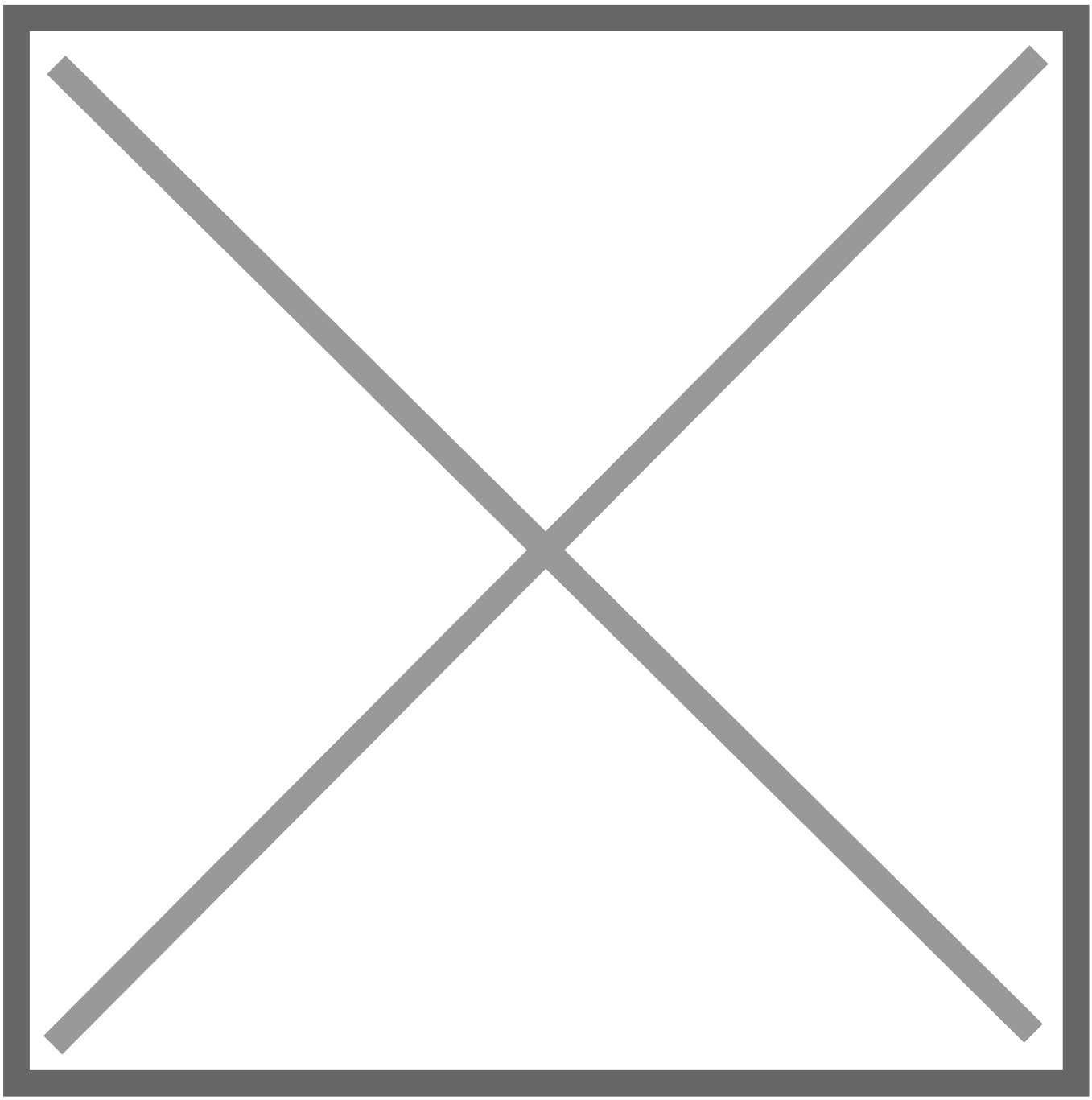
Subscriptions only work for booru-like galleries that put the newest files first, and they only keep up with new content--once they have done their first sync, which usually gets the most recent hundred files or so, they will never reach further into the past. Getting older files, as you will see later, is a job best done with a normal download page.

Here's the dialog, which is under *network->downloaders->manage subscriptions*:

This is a very simple example--there is only one subscription, for safebooru. It has two 'queries' (i.e. searches to keep up with).

It is important to note that while subscriptions can have multiple queries (even hundreds!), they *generally* only work on one site. Expect to create one subscription for safebooru, one for artstation, one for paheal, and so on for every site you care about. Advanced users may be able to think of ways to get around this, but I recommend against it as it throws off some of the internal check timing calculations.

Before we trip over the advanced buttons here, let's zoom in on the actual subscription:



This is a big and powerful panel! I recommend you open the screenshot up in a new browser tab, or in the actual client, so you can refer to it.

Despite all the controls, the basic idea is simple: Up top, I have selected the 'safebooru tag search' download source, and then I have added two artists--"hong_soon-jae" and "houtengeki". These two queries have their own panels for reviewing what URLs they have worked on and further customising their behaviour, but all they *really* are is little bits of search text. When the subscription runs, it will put the given search text into the given download source just as if you were running the regular downloader.

For the most part, all you need to do to set up a good subscription is give it a name, select the download source, and use the 'paste queries' button to paste what you want

to search. Subscriptions have great default options for almost all query types, so you don't have to go any deeper than that to get started.

Do not change the max number of new files options until you know *exactly* what they do and have a good reason to alter them!

how do subscriptions work?

Once you hit ok on the main subscription dialog, the subscription system should immediately come alive. If any queries are due for a 'check', they will perform their search and look for new files (i.e. URLs it has not seen before). Once that is finished, the file download queue will be worked through as normal. Typically, the sub will make a popup like this while it works:

The initial sync can sometimes take a few minutes, but after that, each query usually only needs thirty seconds' work every few days. If you leave your client on in the background, you'll rarely see them. If they ever get in your way, don't be afraid to click their little cancel button or call a global halt with *network->pause->subscriptions*--the next time they run, they will resume from where they were before.

Similarly, the initial sync may produce a hundred files, but subsequent runs are likely to only produce one to ten. If a subscription comes across a lot of big files at once, it may not download them all in one go--but give it time, and it will catch back up before you know it.

When it is done, it leaves a little popup button that will open a new page for you:

This can often be a nice surprise!

what makes a good subscription?

The same rules as for downloaders apply: **start slow, be hesitant, and plan for the long-term.** Artist queries make great subscriptions as they update reliably but not too often and have very stable quality. Pick the artists you like most, see where their stuff is posted, and set up your subs

like that.

Series and character subscriptions are sometimes valuable, but they can be difficult to keep up with and have highly variable quality. It is not uncommon for users to only keep 15% of what a character sub produces. I do not recommend them for anything but your waifu.

Attribute subscriptions like 'blue_eyes' or 'smile' make for terrible subs as the quality is all over the place and you will be inundated by too much content. The only exceptions are for specific, low-count searches that really matter to you, like 'contrapposto' or 'gothic trap thighhighs'.

If you end up subscribing to eight hundred things and get ten thousand new files a week, you made a mistake. Subscriptions are for *keeping up* with things you like. If you let them overwhelm you, you'll resent them.

Subscriptions syncs are somewhat fragile. Do not try to play with the limits or checker options to download a whole 5,000 file query in one go--if you want everything for a query, run it in the manual downloader and get everything, then set up a normal sub for new stuff. There is no benefit to having a 'large' subscription, and it will trim itself down in time anyway.

It is a good idea to run a 'full' download for a search before you set up a subscription. As well as making sure you have the exact right query text and that you have everything ever posted (beyond the 100 files deep a sub will typically look), it saves the bulk of the work (and waiting on bandwidth) for the manual downloader, where it belongs. When a new subscription picks up off a freshly completed download queue, its initial subscription sync only takes thirty seconds since its initial URLs are those that were already processed by the manual downloader. I recommend you stack artist searches up in the manual downloader using 'no limit' file limit, and when they are all finished, select them in the list and *right-click->copy queries*, which will put the search texts in your clipboard, newline-separated. This list can be pasted into the subscription dialog in one go with the 'paste queries' button again!

The entire subscription system assumes the source is a typical 'newest first' booru-style search. If you dick around with some order_by:rating/random metatag, it will not work reliably.

how often do subscriptions check?

Hydrus subscriptions use the same variable-rate checking system as its thread watchers, just on a larger timescale. If you subscribe to a busy feed, it might check for new files once a day, but if you enter an artist who rarely posts, it might only check once every month. You don't have to do anything. The fine details of this are governed by the 'checker options' button. **This is one of the things you should not mess with as you start out.**

If a query goes too 'slow' (typically, this means no new files for 180 days), it will be marked DEAD in the same way a thread will, and it will not be checked again. You will get a little popup when this happens. This is all editable as you get a better feel for the system--if you wish, it is completely

possible to set up a sub that never dies and only checks once a year.

I do not recommend setting up a sub that needs to check more than once a day. Any search that is producing that many files is probably a bad fit for a subscription. **Subscriptions are for lightweight searches that are updated every now and then.**

(you might like to come back to this point once you have tried subs for a week or so and want to refine your workflow)

---

ok, I set up three hundred queries, and now these popup buttons are a hassle

One the edit subscription panel, the 'presentation' options let you publish files to a page. The page will have the subscription's name, just like the button makes, but it cuts out the middle-man and 'locks it in' more than the button, which will be forgotten if you restart the client. **Also, if a page with that name already exists, the new files will be appended to it, just like a normal import page!** I strongly recommend moving to this once you have several subs going. Make a 'page of pages' called 'subs' and put all your subscription landing pages in there, and then you can check it whenever is convenient.

If you discover your subscription workflow tends to be the same for each sub, you can also customise the publication 'label' used. If multiple subs all publish to the 'nsfw subs' label, they will all end up on the same 'nsfw subs' popup button or landing page. Sending multiple subscriptions' import streams into just one or two locations like this can be great.

You can also hide the main working popup. I don't recommend this unless you are really having a problem with it, since it is useful to have that 'active' feedback if something goes wrong.

Note that subscription file import options will, by default, only present 'new' files. Anything already in the db will still be recorded in the internal import cache and used to calculate next check times and so on, but it won't clutter your import stream. This is different to the default for all the other importers, but when you are ready to enter the ranks of the Patricians, you will know to edit your 'loud' default file import options under *options->importing* to behave this way as well. Efficient workflows only care about new files.

how exactly does the sync work?

Figuring out when a repeating search has 'caught up' can be a tricky problem to solve. It sounds simple, but unusual situations like 'a file got tagged late, so it inserted deeper than it ideally should in the gallery search' or 'the website changed its URL format completely, help' can cause problems. Subscriptions are automatic systems, so they tend to be a bit more careful and paranoid about problems, lest they burn 10GB on 10,000 unexpected diaperfur images.

The initial sync is simple. It does a regular search, stopping if it reaches the 'initial file limit' or the last file in the gallery, whichever comes first. The default initial file sync is 100, which is a great number for almost all situations.

Subsequent syncs are more complicated. It ideally 'stops' searching when it reaches files it saw in a previous sync, but if it comes across new files mixed in with the old, it will search a bit deeper. It is not foolproof, and if a file gets tagged very late and ends up a hundred deep in the search, it will probably be missed. There is no good and computationally cheap way at present to resolve this problem, but thankfully it is rare.

Remember that an important 'staying sane' philosophy of downloading and subscriptions is to focus on dealing with the 99.5% you have before worrying about the 0.5% you do not.

The amount of time between syncs is calculated by the checker options. Based on the timestamps attached to existing urls in the subscription cache (either added time, or the post time as parsed from the url), the sub estimates how long it will be before n new files appear, and then next check is scheduled for then. Unless you know what you are doing, checker options, like file limits, are best left alone. A subscription will naturally adapt its checking speed to the file 'velocity' of the source, and there is usually very little benefit to trying to force a sub to check at a radically different speed.

If you want to force your subs to run at the same time, say every evening, it is easier to just use *network->pause->subscriptions* as a manual master on/off control. The ones that are due will catch up together, the ones that aren't won't waste your time.

Remember that subscriptions only keep up with new content. They cannot search backwards in time in order to 'fill out' a search, nor can they fill in gaps. Do not change the file limits or check times to try to make this happen. If you want to ensure complete sync with all existing content for a particular search, use the manual downloader.

In practice, most subs only need to check the first page of a gallery since only the first two or three urls are new.

periodic file limit exceeded

If, during a regular sync, the sub keeps finding new URLs, never hitting a block of already-seen URLs, it will stop upon hitting its 'periodic file limit', which is also usually 100. When it happens, you will get a popup message notification. There are two typical reasons for this:

- A user suddenly posted a large number of files to the site for that query. This sometimes happens with CG gallery spam.
- The website changed their URL format.

The first case is a natural accident of statistics. The subscription now has a 'gap' in its sync. If you want to get what you missed, you can try to fill in the gap with a manual downloader page. Just download to 200 files or so, and the downloader will work quickly to one-time work through the URLs in the gap.

The second case is a safety stopgap for hydrus. If a site decides to have /post/123456 style URLs instead of post.php?id=123456 style, hydrus will suddenly see those as entirely 'new' URLs. It could also be because of an updated downloader, which pulls URLs in API format or similar. This is again thankfully quite rare, but it triggers several problems--the associated downloader usually breaks, as it does not yet recognise those new URLs, and all your subs for that site will parse through and hit the periodic limit for every query. When this happens, you'll usually get several periodic limit popups at once, and you may need to update your downloader. If you know the person who wrote the original downloader, they'll likely want to know about the problem, or may already have a fix sorted. It is often a good idea to pause the affected subs until you have it figured out and working in a normal gallery downloader page.

I put character queries in my artist sub, and now things are all mixed up

On the main subscription dialog, there are 'merge' and 'separate' buttons. These are powerful, but they will walk you through the process of pulling queries out of a sub and merging them back into a different one. Only subs that use the same download source can be merged. Give them a go, and if it all goes wrong, just hit the cancel button on the dialog.

Filtering Duplicates

duplicates

As files are shared on the internet, they are often resized, cropped, converted to a different format, altered by the original or a new artist, or turned into a template and reinterpreted over and over and over. Even if you have a very restrictive importing workflow, your client is almost certainly going to get some **duplicates**. Some will be interesting alternate versions that you want to keep, and others will be thumbnails and other low-quality garbage you accidentally imported and would rather delete. Along the way, it would be nice to merge your ratings and tags to the better files so you don't lose any work.

Finding and processing duplicates within a large collection is impossible to do by hand, so I have written a system to do the heavy lifting for you. It currently works on still images, but an extension for gifs and video is planned.

Hydrus finds *potential* duplicates using a search algorithm that compares images by their shape. Once these pairs of potentials are found, they are presented to you through a filter like the archive/delete filter to determine their exact relationship and if you want to make a further action, such as deleting the 'worse' file of a pair. All of your decisions build up in the database to form logically consistent groups of duplicates and 'alternate' relationships that can be used to infer future information. For instance, if you say that file A is a duplicate of B and B is a duplicate of C, A and C are automatically recognised as duplicates as well.

This all starts on--

the duplicates processing page

On the normal 'new page' selection window, hit *special->duplicates processing*. This will open this page:

Let's go to the preparation page first:

The 'similar shape' algorithm works on *distance*. Two files with 0 distance are likely exact matches, such as resizes of the same file or lower/higher quality jpegs, whereas those with distance 4 tend to be to be hairstyle or costume changes. You will be starting on distance 0 and not expect to ever go above 4 or 8 or so. Going too high increases the danger of being overwhelmed by false positives.

If you are interested, the current version of this system uses a 64-bit phash to represent the image shape and a VPTree to search different files' phashes' relative hamming distance. I expect to extend it in future with multiple phash generation (flips, rotations, and 'interesting' image crops and video frames) and most-common colour comparisons.

Searching for duplicates is fairly fast per file, but with a large client with hundreds of thousands of files, the total CPU time adds up. You can do a little manual searching if you like, but once you are all settled here, I recommend you hit the cog icon on the preparation page and let hydrus do this page's catch-up search work in your regular maintenance time. It'll swiftly catch up and keep you up to date without you even thinking about it.

Start searching on the 'exact match' search distance of 0. It is generally easier and more valuable to get exact duplicates out of the way first.

Once you have some files searched, you should see a potential pair count appear in the 'filtering' page.

the filtering page

Processing duplicates can be real trudge-work if you do not set up a workflow you enjoy. It is a little slower than the archive/delete filter, and sometimes takes a bit more cognitive work. For many users, it is a good task to do while listening to a podcast or having a video going on another screen.

If you have a client with tens of thousands of files, you will likely have thousands of potential pairs. This can be intimidating, but do not worry--due to the A, B, C logical inferrences as above, you will not have to go through every single one. The more information you put into the system, the faster the number will drop.

The filter has a regular file search interface attached. As you can see, it defaults to *system:everything*, but you can limit what files you will be working on simply by adding new search predicates. You might like to only work on files in your archive (i.e. that you know you care about to begin with), for instance. You can choose whether both files of the pair should match the search, or just one. 'creator:' tags work very well at cutting the search domain to something more manageable and consistent--try your favourite creator!

If you would like an example from the current search domain, hit the 'show some random potential pairs' button, and it will show two or more files that seem related. It is often interesting and surprising to see what it finds! The action buttons below allow for quick processing of these pairs and groups when convenient (particularly for large cg sets with 100+ alternates), but I recommend you leave these alone until you know the system better.

When you are ready, launch the filter.

the duplicates filter

We have not set up your duplicate 'merge' options yet, so do not get too into this. For this first time, just poke around, make some pretend choices, and then cancel out and choose to forget them.

Like the archive/delete filter, this uses quick mouse-clicks, keyboard shortcuts, or button clicks to action pairs. It presents two files at a time, labelled A and B, which you can quickly switch between just as in the normal media viewer. As soon as you action them, the next pair is shown. The two files will have their current zoom-size locked so they stay the same size (and in the same position) as you switch between them. Scroll your mouse wheel a couple of times and see if any obvious differences stand out.

Please note the hydrus media viewer does not currently work well with large resolutions at high zoom (it gets laggy and may have memory issues). Don't zoom in to 1600% and try to look at jpeg artifact differences on very large files, as this is simply not well supported yet.

The hover window on the right also presents a number of 'comparison statements' to help you make your decision. Green statements mean this current file is probably 'better', and red the opposite. Larger, older, higher-quality, more-tagged files are generally considered better. These statements have scores associated with them (which you can edit in *file->options->duplicates*), and the file of the pair with the highest score is presented first. If the files are duplicates, you can *generally* assume the first file you see, the 'A', is the better, particularly if there are several green statements.

The filter will need to occasionally checkpoint, saving the decisions so far to the database, before it can fetch the next batch. This allows it to apply inferred information from your current batch and reduce your pending count faster before serving up the next set. It will present you with a quick interstitial 'confirm/back' dialog just to let you know. This happens more often as the potential count decreases.

the decisions to make

There are three ways a file can be related to another in the current duplicates system: duplicates, alternates, or false positive (not related).

False positive (not related) is the easiest. You will not see completely unrelated pairs presented very often in the filter, particularly at low search distances, but if the shape of face and hair and clothing happen to line up (or geometric shapes, often), the search system may make a false positive match. In this case, just click 'they are not related'.

Alternate relations are files that are not duplicates but obviously related in some way. Perhaps a costume change or a recolour. Hydrus does not have rich alternate support yet (but it is planned, and highly requested), so this relationship is mostly a 'holding area' for files that we will revisit for further processing in the future.

Duplicate files are of **the exact same thing**. They may be different resolutions, file formats, encoding quality, or one might even have watermark, but they are fundamentally different views on the exact same art. As you can see with the buttons, you can select one file as the 'better' or say they are about the same. If the files are basically the same, there is no point stressing about which is 0.2% better--just click 'they are the same'. For better/worse pairs, you might have reason to keep both, but most of the time I recommend you delete the worse.

You can customise the shortcuts under *file->shortcuts->duplicate_filter*. The defaults are:

- Left-click or space: **this is better, delete the other**.
- Right-click: **they are related alternates**.
- Middle-click: **Go back one decision.**
- Enter/Escape: **Stop filtering.**

merging metadata

If two duplicates have different metadata like tags or archive status, you probably want to merge them. Cancel out of the filter and click the 'edit default duplicate metadata merge options' button:

By default, these options are fairly empty. You will have to set up what you want based on your services and preferences. Setting a simple 'copy all tags' is generally a good idea, and like/dislike ratings also often make sense. The settings for better and same quality should probably be similar, but it depends on your situation.

If you choose the 'custom action' in the duplicate filter, you will be presented with a fresh 'edit duplicate merge options' panel for the action you select and can customise the merge specifically for that choice. ('favourite' options will come here in the future!)

Once you are all set up here, you can dive into the duplicate filter. Please let me know how you get on with it!

what now?

The duplicate system is still incomplete. Now the db side is solid, the UI needs to catch up. Future versions will show duplicate information on thumbnails and the media viewer and allow quick-navigation to a file's duplicates and alternates.

For now, if you wish to see a file's duplicates, right-click it and select *file relationships*. You can review all its current duplicates, open them in a new page, appoint the new 'best file' of a duplicate group, and even mass-action selections of thumbnails.

You can also search for files based on the number of file relations they have (including when setting the search domain of the duplicate filter!) using *system:file relationships*. You can also search for best/not best files of groups, which makes it easy, for instance, to find all the spare duplicate files if you decide you no longer want to keep them.

I expect future versions of the system to also auto-resolve easy duplicate pairs, such as clearing out pixel-for-pixel png versions of jpgs.

game cgs

If you import a lot of game CGs, which frequently have dozens or hundreds of alternates, I recommend you set them as alternates by selecting them all and setting the status through the thumbnail right-click menu. The duplicate filter, being limited to pairs, needs to compare all new members of an alternate group to all other members once to verify they are not duplicates. This is not a big deal for alternates with three or four members, but game CGs provide an overwhelming edge case. Setting a group of thumbnails as alternate 'fixes' their alternate status immediately, discounting the possibility of any internate duplicates, and provides an easy way out of this situation.

more information and examples

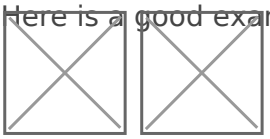
better/worse

Which of two files is better? Here are some common reasons:

- higher resolution
- better image quality
- png over jpg for screenshots
- jpg over png for busy images
- jpg over png for pixel-for-pixel duplicates
- a better crop
- no watermark or site-frame or undesired blemish
- has been tagged by other people, so is likely to be the more 'popular'

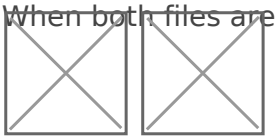
However these are not hard rules--sometimes a file has a larger resolution or filesize due to a bad upscaling or encoding decision by the person who 'reinterpreted' it. You really have to look at it and decide for yourself.

Here is a good example of a better/worse pair:



The first image is better because it is a png (pixel-perfect pngs are always better than jpgs for screenshots of applications--note how obvious the jpg's encoding artifacts are on the flat colour background) and it has a slightly higher (original) resolution, making it less blurry. I presume the second went through some FunnyJunk-tier trash meme site to get automatically cropped to 960px height and converted to the significantly smaller jpeg. Whatever happened, let's drop the second and keep the first.

When both files are jpgs, differences in quality are very common and often significant:

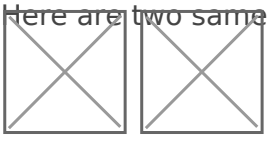


Again, this is mostly due to some online service resizing and lowering quality to ease on their bandwidth costs. There is usually no reason to keep the lower quality version.

same quality duplicates

When are two files the same quality? A good rule of thumb is if you scroll between them and see no obvious differences, and the comparison statements do not suggest anything significant, just set them as same quality.

Here are two same quality duplicates:



There is no obvious different between those two. The filesize is significantly different, so I suspect the smaller is a lossless png optimisation, but in the grand scheme of things, that doesn't matter so much. Many of the big content providers--Facebook, Google, Cloudflare--automatically 'optimise' the data that goes through their networks in order to save bandwidth. Although jpegs are often a slaughterhouse, with pngs it is usually harmless.
Given the filesize, you might decide that these are actually a better/worse pair--but if the larger image had tags and was the 'canonical' version on most boorus, the decision might not be so clear. You can choose better/worse and delete one randomly, but sometimes you may just want to keep both without a firm decision on which is best, so just set 'same quality' and move on. Your time is more valuable than a few dozen KB.
Sometimes, you will see pixel-for-pixel duplicate jpegs of very slightly different size, such as 787KB vs 779KB. The smaller of these is usually an exact duplicate that has had its internal metadata (e.g. EXIF tags) stripped by a program or website CDN. They are same quality unless you have a strong opinion on whether having internal metadata in a file is useful.

- # alternates

As I wrote above, hydrus's alternates system in not yet properly ready. It is important to have a basic 'alternates' relationship for now, but it is a holding area until we have a workflow to apply 'WIP'- or 'recolour'-type labels and present that information nicely in the media viewer.
Alternates are not of exactly the same thing, but one is variant of the other or they are both descended from a common original. The precise definition is up to you, but it generally means something like:

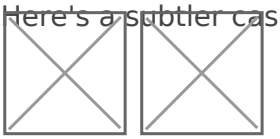
- the files are recolours
- the files are alternate versions of the same image produced by the same or different artists (e.g. clean/messy or with/without hair ribbon)
- iterations on a close template
- different versions of a file's progress, such as the steps from the initial draft sketch to a final shaded version

Here are some recolours of the same image:

And some WIP:

And a costume change:

None of these are duplicates, but they are obviously related. The duplicate search will notice they are similar, so we should let the client know they are 'alternate'.
Here's a subtler case:

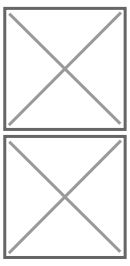


These two files are very similar, but try opening both in separate tabs and then flicking back and forth: the second's glove-string is further into the mouth and has improved chin shading, a more refined eye shape, and shaved pubic hair. It is simple to spot these differences in the client's duplicate filter when you scroll back and forth.
I believe the second is an improvement on the first by the same artist, so it is a WIP alternate. You might also consider it a 'better' improvement.
Here are three files you might or might not consider to be alternates:

These are all based on the same template--which is why the dupe filter found them--but they are not so closely related as those above, and the last one is joking about a different ideology entirely and might deserve to be in its own group. Ultimately, you might prefer just to give them some shared tag and consider them not alternates *per se*.

- # not related/false positive

Here are two files that match false positively:



Despite their similar shape, they are neither duplicates nor of even the same topic. The only commonality is the medium. I would not consider them close enough to be alternates--just adding something like 'screenshot' and 'imageboard' as tags to both is probably the closest connection they have.

Recording the 'false positive' relationship is important to make sure the comparison does not come up again in the duplicate filter.
The incidence of false positives increases as you broaden the search distance--the less precise your search, the less likely it is to be correct. At distance 14, these files all match, but uselessly:

the duplicates system

(advanced nonsense, you can skip this section. tl;dr: duplicate file groups keep track of their best quality file, sometimes called the King)

Hydrus achieves duplicate transitivity by treating duplicate files as groups. Although you action pairs, if you set (A duplicate B), that creates a group (A,B). Subsequently setting (B duplicate C) extends the group to be (A,B,C), and so (A duplicate C) is transitively implied.

The first version of the duplicate system attempted to record better/worse/same information for all files in a virtual duplicate group, but this proved very complicated, workflow-heavy, and not particularly useful. The new system instead appoints a single *King* as the best file of a group. All other files in the group are beneath the King and have no other relationship data retained.

This King represents the group in the duplicate filter (and in potential pairs, which are actually recorded between duplicate media groups--even if most of them at the outset only have one member). If the other file in a pair is considered better, it becomes the new King, but if it is worse or equal, it merges into the other members. When two Kings are compared, whole groups can merge!

Alternates are stored in a similar way, except the members are duplicate groups rather than individual files and they have no significant internal relationship metadata yet. If α, β, and γ are duplicate groups that each have one or more files, then setting (α alt β) and (β alt γ) creates an alternate group (α,β,γ), with the caveat that α and γ will still be sent to the duplicate filter once just to check they are not duplicates by chance. The specific file members of these groups, A, B, C and so on, inherit the relationships of their parent groups when you right-click on their thumbnails.

False positive relationships are stored between pairs of alternate groups, so they apply transitively between all the files of either side's alternate group. If (α alt β) and (ψ alt ω) and you apply (α fp ψ), then (α fp ω), (β fp ψ), and (β fp ω) are all transitively implied.

Some fun. And simpler.

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.