Changelog <49

- ## version 49

    - local rating like service
    - local rating numerical service
    - ratings dialog
    - ratings drawn on preview and fullscreen canvas
    - search by rating
    - system:rating in options
    - sort by rating
    - a ratings help page added
    - custom shortcuts
    - overhauled shortcuts code
    - a new, comprehensive regex menu in the regex tag dialog
    - extensive new hentai foundry search and sort options
    - a new option regarding the search style of tag A/C entries
    - thu dumper can now dump with tags
    - regex tag dialog selects tag A/C on page change
    - regex tag dialog's button now now shows in import dialog even if there are no remote tag dialogs with POST privilege
    - A/C is slightly faster in all cases
    - preview canvas now updates on a content update
    - numtags = 0 and numtags > 0 now work for pending tags
    - shift- or Ctrl-Click no longer deselects
    - listbook renaming and resizing bug fixed
    - collections thumbnail is fixed
    - sometimes, an empty tag, '', could be entered into pending and screw a lot up
    - sometimes, a null tag could be entered into pending and screw a lot up
    - advanced tag options has better checkbox defaults
    - some help screenshots updated
    - the db's processing is slightly rebalanced
    - the db's startup is slightly rebalanced
    - a misc gui bug was fixed
    - a misc gui typo was fixed

- # version 48

  - New copy path and copy url menu entries on thumbnail right-click menu
  - Gif scrollbar
  - Hentai Foundry parsing errors fixed
  - Hentai Foundry 404 errors fixed
  - All the html parsers are rewritten using BeautifulSoap
  - Flash now zooms full, but not fit
  - Filter is now F12
  - Help is updated in a number of ways
  - Up/Down on empty input to change repository in manage tags dialog is fixed
  - The servers now support OPTIONS requests
  - Several status typos fixed
  - I started local ratings services

- # version 47

  - Hentai Foundry interface is done, for both artist and tag searches
  - A/C gui reorganisation
  - A/C file and repo buttons added
  - A/C cache count alterations to match those buttons
  - A/C tag processing is cleaned up
  - Search and thumbnail-action improved to support null file service or non-null tag service
  - system:numtags=0 and system:numtags>0 optimised to be much faster
  - Ctrl+I now switches searching immediately/waiting
  - z now switches between zoom full and zoom fit in fullscreen view
  - advanced import options now has exclude already deleted files checkbox
  - Thumbnails choose the way they access the db more intelligently
  - ManagementPanel colour is changed, hmmm
  - Some DB cleanup
  - Some general code cleanup
  - Some download logic improved, including timer
  - Some download status improved
  - Download 404 notices better
  - DeviantArt now initially shows 'artist username' in its searchbox
  - Some code nomenclature improved
  - I've made a changelog.html, compling all changes together.

- # version 46

  - DA parsing CLEAR
  - rule34@paheal parsing CLEAR
  - tbib (another booru) parsing CLEAR
  - Canvas rearranging
  - Preview now shows tags
  - The autocomplete tag write dialog will promote existing exact matches to the top of the list, even if it is not the most popular.
  - Autocomplete lists now scroll with the mouse.
  - A/C scrolling is a little more accurate.
  - Pressing up or down on an empty A/C entry will change tag repository.
  - A sledgehammer has been applied to the python garbage collector.
  - The code that navigates boorus is better.
  - The import queuing/pausing/cancelling system is better.
  - Booru tag-processing is better.
  - The import-queue gui is slightly better sized on small screens.
  - The new local tags service now shows up in advanced tag options.
  - Even more flicker is removed from the main thumbnail view, this time when clicking on whitespace. (it was a stupid bug)
  - Serverside db-backup now includes the WAL file.
  - Manually adding an admin service is fixed.

- # version 45

  - Local tags: DONE
  - Thread watcher: DONE
  - Fullscreen now shows tags
  - Preview shows some more file + tag info
  - Listbooks load their panels on demand.
  - Local options dialog is now a listbook.
  - Some incorrect 'sample image' downloads from danbooru are fixed. Danbooru's inner workings remain a 404-cluttered mess though.
  - Some remaining utf-8 html parse errors (where a limited content-type header is given) are cleared up.
  - Another connection problem (a certain case of the server closing the connection) is now properly recovered from.
  - Some flicker in the thumbnail view is reduced, but my slow laptop still shows some, so I need to revisit this.

- Autocomplete tag boxes now have little tooltips better describing their search domains.

version 44

- All database triggers are removed and replaced with more efficient inline code.
- Cached number of files, namespaces and tags for a tag repo are kept updated, so review services should rarely take very long to load.
- Cached number of A/C counts (current and pending) are kept updated, so A/C results should rarely take very long to load.
- The way A/C cache searches are done is improved.
- The way A/C cache entries are calculated is vastly improved.
- The way several db SELECT DISTINCT * and SELECT COUNT(DISTINCT *) queries are performed is improved.
- The custom httpconnection object is improved to better deal with server-side connection closes. (Cannot connect to server problem)
- The custom httpconnection object does http redirections more intelligently.
- The custom httpconnection object now parses non-ascii content-type (e.g. utf-8) properly, according to server-side header. The tag "pokémon" will now parse correctly.
- e621 and rule34@booru.org are fixed. (just a couple parsing-variable changes)
- If some tag parsing entries are checked on a booru download page, the tags will be downloaded, parsed and applied even to files already in the database.
- The mouse cursor will remain visible in the fullscreen browser while the right click menu is open.
- I corrected some which->that grammar in a couple dialogs.
- Auto-repo-setup (from the help menu) was broken in a very stupid way, but I fixed it. I apologise for not noticing.
- Thumbnails now show creator and title up top. They are a bit crushed, so I may rearrange them.

version 43

- Updated import + booru code
- New booru buttons
- Cut tag parse bug fixed
- Download preview instead of real file bug fixed (I think)
- Fixed the 'thumbnails not appearing on import' bug
- Import flickers less

- Media Classes rewrite is done, although I'm not totally happy. Have to think about it.
- A problem with pending tags applying over existing tags no flushing correctly is fixed.
- A/C cache is better, but still a little slow on slow computers.
- system:duration=0 is fixed.

version 42

- Autocomplete tag cache DONE. Please tell me if you encounter excessive slowdown or incorrect counts.
- Autocomplete tag logic improved in about five ways. Tags should appear more intuitively and with more accurate counts.
- Copy and paste buttons in manage tag dialog to facilitate spamming many tags from one file to another.
- Imageboard dump is fixed!
- URL download got fixed last week and I forgot to say!
- A peculiar pseudo-foreign-key bug with the active_pending_mappings table is sorted. (leading to more accurate queries and counts)
- Since display of large queries is so much faster, the 'woah! 12,000 query results' warning is gone.
- Fullscreen precache timings are rebalanced, hopefully for the better!
- The importing gui classes are a little smoother, and thumbs should load one by one, not row by row, once the page height is exceeded. The scrollbars should be less janky.
- Page transitions are smoother.
- Importing from URL will now load those urls that are already inside the db (hella fast, too), just like importing from hdd.
- The double-update-download bug is finally fixed! Now you'll download update number 258 only once, and not twice in a row. (I hope!) It never mattered much, beyond a little CPU, but it annoyed the hell out of me. Working out what was wrong took ages. (Long story short: Python's Priority Queues are not parallel FIFOs!)
- There was a weird bug in pending tags sometimes going to higher tag-precedence services, rather than the one chosen in the manage tags dialog.
- There was a typo in reset service.
- More content update unification in the db, making things much neater behind the scenes.
- I found a potential hang in the db-shutdown code.
- I found another potential hang in the db-shutdown code!
- The number of messages in your inbox is now displayed on the statusbar.

- Bold, Italic and Underline now work with Ctrl+B/I/U in the message compose panel.
- The 'don't receive messages from Anonymous' checkbox in Add, Remove and Edit Services for message depots now does something.
- system:unread in message search now has a count.
- A draft will remember its saved changes when you click on another conversation and back again.
- You can now save a draft with zero recipients.

version 41

- Rich Text Control with Toolbar in messaging.
- Message unread/read status now syncs to message depot.
- Can set defaults for file system predicates.
- Animated Text class written.
- system:unread tag in messaging.
- Import bug fixed.
- Messaging search logic improved.
- Message depot default check time down to 180, and daemon thread improved to manage with that.
- Application startup is more stable.
- Some AddMessage logic improved.
- Drag+Drop service fixed and improved.
- Tags added while importing will now turn up in autocomplete.
- I unified more of the client's content-update db calls. Just making stuff neater.
- I massively simplified the autocomplete tag code in preparation for another bash at doing autocomplete cache. A/C is far too slow atm, imo.
- Some A/C counts are still off, particularly when there are a mix of current and pending. I couldn't figure out what is causing it, but I'll look at it more as I do the cache.
- In add, remove and edit serivces, you can now specify if you want to receive messages from Anonymous or not. Unfortunately, because I forgot to implement it in DoMessageQuery, this checkbox doesn't actually do anything yet! Whoops!

version 40

- Inbox/Archive for messaging works. The inbox icon is back, and F7 to archive as usual.

- Unread/Read for messaging works. Click on your status next to a message panel to change.
- All message participants are shown in the conversation listctrl.
- The message page will open with 'system:inbox' active, ordered by newest message.
- I fixed conversation listctrl sorting in a couple ways.
- Some bugs in the db->gui update reporting system are fixed, so the compose window and draft panel will close properly on a message-send, and the conversation listctrl will swiftly update unread counts and inbox status and so on.
- The draft panel's save button wasn't updating sometimes when it should have been.
- html links in messages will show in the main frame's statusbar and launch in your browser when clicked.
- I reworked how many system tags are searched in file queries to make them muuuuch faster. Now, any time you go system:hash, :size, :mime, :age, :width, :height, :duration, it (should) be significantly faster than before.
- Ever have a png file that had an ok thumbnail but a black preview/fullscreen? It was probably an 'LA' image, or 'greyscale with an alpha channel'. Display of LA images is now fixed.
- Streamlined the way file 'archive' and 'delete' commands are sent throughout the application, with an eye to eventually adding all other user-commands to the same system (to make it simpler, and also to do undo-redo).
- I've updated the help docs, including the section on how messaging actually works.
- You can now enter system tags with enter/return from the keyboard, not just double-clicking with the mouse.

- ## version 39

  - A whole bunch of stuff I didn't write down that generally made messages work beyond sending "Hello World".
  - Unicode messages now work.
  - Newlines in messages now display properly.
  - Fixed contact saving and drag-and-drop to the manage contacts dialog.
  - You can now add a contact using a contact address (assuming their service is v39).
  - A bug in contact renaming is fixed.
  - My contact (hydrus admin) is now added automatically.
  - Messages are orphan-deleted properly server-side.
  - Message depots you control now get an entry under the admin menu.
  - The conversation listctrl is now properly sorted.

- The HtmlWindow that shows message body is now resized to its minimum needed height via magic that I ain't gotta explain.
- Messages that fail to send can be retried by clicking on the failed.
- Some draft save/send button enable/disable logic was updated.
- I switched around some server-side account logic.
- You can't zoom a flash window bigger than the screen.
- I've made the mouse appear in fullscreen when you move it, and this includes while you drag. See how you feel about it. I'll make options to change behaviour if people care.
- Better zip structure. (No more CWD rubbish)
- New tar.gz release with only the code and help. Code is 1.2MB, help is 14MB lol. I'll improve the structure of this release next week. I threw it together by hand today, but I'll improve its structure next week.

- # version 38

  - messaging v1.0

- # version 37

  - messaging prototype

- # version 36

  - animated gif fixes
  - new page chooser
  - svcp->cstvcp in fullscreen too
  - new import progress
  - just start the server on this computer, please
  - I think I have scrolling to the current selection after an archive fixed now.
  - Started an experimental simpler way to send commands to the db, which may open the way for undo-redo.
  - Capitalisation option in file->options->gui. It only does the menubar for now, but I will expand it.
  - Custom sorts/collects. Under file->options. Check it out, play with it if you care about it.

- Some program initialisation error handling is cleaned up.
- Can now no longer start a server if the admin port it wants to use is occupied. (can't run two servers at once on the same port)
- I broke dump with a typo in a previous release, I think! It is fixed now.

version 35

- Slight speedup of review services frame loadtime.
- SVCP now replaced with SCTVCP and CSTVCP for now, with more planned next week.
- Repo sync daemon adds db jobs less manically, now, locking up the gui less.
- Repo sync daemon reports successful updates to the log.
- Tag petition review page has a visual bug fixed.
- ThumbnailMediaPanel manages its underlying bmp canvas more intelligently.
- Some ugly page-closing code was cleaned up.
- There is a 'regex examples' popout window on the regex-filename-tag import dialog, with copy-pastable examples of python regexes that'll work for common filename-parsing situations.
- Home and End keys now select the first/last thumbnail as well as scrolling.

version 34

- DB consolidation, and foreign keys and triggers with it. The db is less prone to orphan ids and miscounts.
- For long-time users, all orphan ids should have all been deleted, reducing weirdness.
- Total size of local service is now tracked accurately.
- Archive while importing should work a bit better now.
- Update logic should be more reliable, as should all service_updates.
- Archive count for file repos should be less wrong.
- Fixed two (!) typos in system:age predicate code.
- File upload is fixed (not sure if it was ever broken in release, but I fixed the stupid thing I did anyway)
- DB (and server) errors should report better now, revealing inner-db code in the traceback. (and printing it to the appropriate log)
- Improved the mime_to_enum object for image/jpeg, image/jpg, image/jpe, image/wtfaretheygoingtocalljpegnow.
- HTTP connections can deal with some esoteric commands, like 301, without throwing a fit. All http goes through the same code.

- Danbooru should now be in your booru list, and it _should_ work.
- Pausing/Cancelling a booru gallery parse should now work.
- 4chan dumper now understands captchas being wrong and the thread not existing; in the former case it'll let you put in a new captcha and try again, in the latter it'll stop the whole dump.
- Some general code neatness improvements.

· version 33

- Boorus are working for users who started last week
- The server can now manage many more simultaneous requests without bugging out.
- The early freeze/import freeze due to simultaneous access in the client is fixed.
- The client's file+thumbnail server (details in help's advanced.html) works again.
- Client DB now stores md5 and sha1 for all local files.
- Client DB has a larger cache size.
- The freeze caused when importing thousands of files at once with tags is fixed.
- The yaml imageboard import (when you drag and drop a .yaml file onto the imageboards dialog) now adds form fields correctly! If you had problems here, check out install_path/static/default_imageboards.yaml to reset them!
- The autocomplete counts should be more accurate, particularly when there are a mix of current and pending tags in the db.
- The autocomplete in the regex-path dialog will now search the db correctly (it was misconfigured).
- Autocomplete is also just a _little_ faster
- File imports are now low priority, so gui activity should still happen nice and quick even while an import is occurring.
- The review services frame will now show how long until the next update will be checked (see picture above). In case of error, it will show how long until it tries again.
- The review services panel and manage services panel are of a better default size, and they use my new listbook class exclusively, rather than the old notebook.
- I've added some tooltips on not-obvious widgets.
- Content Updates have vastly improved processing speed on a thumbnail view (x100 in some cases!). Tag updates particularly will not be so slow to process if you have a large search open.
- The import gauge will now show the correct range according to the current queue size, not always the default 100.
- Active Import Pages will throw up a yes/no dialog if you try to close them.
- Some beautifying of code syntax. '\n' is now os.linesep, which means exported stuff should show up better in notepad.

- Some changes in network protocol, although now I think of them, I realise I forgot to update the network version. They aren't important for most users, so w/e.

version 32

- client db serialisation DONE
- Cleaned up a dirty-bmp canvas init issue.
- Resized some dlg stuff that was too tall
- Cleaned up a bunch of unrelated db code
- Tightened the pubsub system so it integrates better with wx.

version 31

- booru prototype
- Fixed some problem with last error timestamp
- Fixed a bug in repo bandwidth tracking
- Fixed URL registration with the client db.
- Made downloading URLs asynchronous again
- Made repo synchronisations not block so much, and not for nearly as long. You should be able to do queries and click on thumbs (albeit at a slower rate) while the client is downloading and processing updates.

version 30

- Thumbs are exported, and, along with perceptual hashes, now treated as a global pool, reducing downloads and speeding queries.
- Thumbnail resize is now asynchronous, done as each thumbnail is requested. Changing the thumbnail size in the options dialog happens real quick, and if you don't like a particular size, you can change it back just as fast.
- Temporary DB locking mechanisms ensure serialisation, less inefficient locking.
- ImportPage superclass works a lot better, in a lot less code. DB import code is now even simpler, and much faster, especially for redundant files. The new import page will also tell you how many files were successful/failed/already deleted/already in the db.
- Fetching images for fullscreen or preview is fully asynchronous, and will no longer block the gui thread.

- Some query logic is improved.
- You can now import and export sites/imageboards from the manage imageboards dialog. It'll export to a .yaml file, which is easy to understand with a decent text editor. Just drag and drop onto the dialog to import.
- Selecting thumbs has a little less preview-flicker now.
- The new tagboxes now render numbers prettily, so 4123 -> 4,123.
- (+) namespace:tag now colours properly.
- The background for selected tags is now an appropriate colour.
- You can now choose the colours of any namespace you like in the normal options dialog.
- Clicking in whitespace in the new tagbox deselects, rather than select(0)s.
- Spacebar, PageDown, Return and all the other special command characters are now propagated to flash in fullscreen view, just as long as the mouse is over the canvas.

version 29

- Big changes to the db's file storage, as above.
- database->vacuum is reinstated, database->reindex is retired. I recommend you run a vacuum once you are updated; it'll only take a minute now.
- Rollback and initialisation bugs are fixed.
- There's a splash window now that can offer client-boot feedback.
- The way scrolling-to-focus is calculated is improved.
- Pretty much everywhere you see tags will now show coloured tags! I finished up my own tagbox class, which for now will show tags coloured in the default booru scheme. It also loads in a few milliseconds, rather than the ~0.5s of the default listbox. It scrolls faster, too. I'll have user-customisable colours next release.
- Files with invalid resolution, size and mime are now filtered out before dumping to an imageboard.
- A number of misc other bugs were fixed.

version 28

- I fixed the 'searching immediately' bug. It was a stupid typo as usual.
- Post-Filter scrolling should be a bit more reliable.
- Some more log events.
- I did a lot of work on my new tags box, although the actualy gui elements will have to be finished for v29.
- Manage Imageboards wasn't saving properly. It is now.

- My new progressdialog happens in several places. It is a little neater than the one in wx; just better for my purposes.
- I reworked some of the transaction stuff in the db so that I can atomically queue pubsub events. (this means file import are less juddery)
- After becoming very annoyed with wx's listbook/labelbook/whateverbook, I wrote my own. It's all over the place in my dialogs now. It is tighter, more reliable, and can scroll.
- I reworked some of the way sets interacted.
- There was an infinite recursion bug sometimes, when uploading files to a file repo.

version 27

- Cleared out some old unused library imports, so the client is slightly smaller.
- Had a cyclical import (!). Luckily, python is forgiving, but I have still fixed it.
- Collection thumbnails will now update with their new pending tags correctly.
- Some wx.Panel vs wx.Window issues and nomenclature resolved.
- Full Page decouple. Now the window to the left is a seperate, much neater class. It and the media/preview windows talk to each other in a clever way that means I can make new kinds of page much faster. It scrolls, too.
- Thumbnails resize will tell the thumbnail window just to redraw itself, not refresh the entire query.
- 'system:local' and 'system:not local' are now mutually exclusive (will swap each other out, rather than overlap), like inbox and archive.
- A little neatening of the system predicate code.
- 'system:similar_to' RMB menu entry fixed for some cases.
- There is a new log window. Go view->new log page to see it. It doesn't show much right now, but you can see synchro successes and errors pop up as they happen.
- There is now a popup button on the import dialog and the URL downloader page that does 'advanced import options'. You can now tell these import processes to archive everything and/or ignore files below a certain size or resolution. By default, the client will not import anything <5KB or <50x50px. You can set no limit.
- Flash windows will now own all keystrokes as long as the mouse cursor is over the flash. If you are browsing/filtering, just move your mouse over or off the flash window to make sure your return/page down/spacebar/escape/whatever is interpreted as you want.
- The URL downloader will not download the same file twice in a row, so you can add a thread URL several times in a row, and it'll only download the newest files. The behaviour of this is affected by the 'exclude deleted files' checkbox in the client's options. I'll implement thread watching a bit later.

version 26

- The double unnamespaced count bug ('system:blah (70)', 'blah (140)') in autocomplete should be gone.
- Some rejiggering of how the statusbar updates means active import windows can't spam the statusbar when they don't have focus.
- The count of pending tags is more accurate and uniform. When it says (+3), it means for three files, not in three cases.
- Tag counts overall are better. When it says 'diaper vore (17)', you should be getting 17 files if you enter that search.
- GUI tag repo precedence is slightly more accurate, compared to what the db thinks it should be.
- I nailed down why system:mime wasn't working sometimes, and now it should work all the time.
- I discovered some problems with repo-side bandwidth tracking. I fixed a couple bits, but have notes to look at it more.
- In v25, if system:limit was greater than the search population, it threw an error! Whoops! Fixed now.
- The post-filtering thumbnail problem is totally fixed, as are a couple other instances when thumbs would sometimes not draw.
- The fullscreen change-image-jittering-around problem is fixed, I think.
- Fullscreen archive works again. Not sure why/when it broke.
- Importing with archive checkbox ticked sometimes didn't archive, lol.
- A bunch of decoupling of gui elements, making for much neater code and less gui blocking.
- Decoupling of import from hdd as well.
- Query is now asynchronous! If you do a massive query on a slow computer, you won't lose the gui! You can go check other tabs, whatever, and the query will process in the background, rendering on the page you started it when it is done.
- A number of cpu-heavy numbers are now cleverly cached, so system predicate fetch happens much faster, especially on 100,000+ image dbs.
- Some 'collect' code is sped up and made generally less bonkers.
- I wrote a couple new classes for my most common gui elements.
- There is now an option in file->options to change the number of characters before autocomplete will go search the db for tags! If you have a slow computer and a huge db, you can set it to 2, 3 or even more chars. Default is 1.
- Dialogs that ask you questions will now usually center on their parent window.
- The current selection tagsbox won't show ('blah (12)', 'blah (+3)'), it'll combine into 'blah (12) (+3) (-1)'. With the + meaning pending and the - meaning petitioned.
- Autocomplete now respects which of the include_current and include_pending buttons are green! It'll only search current/pending as you wish! And the

numbers will now be accurate!
- My new question dialog with the yes/no buttons hooks the Escape key to mean no.
- I don't use it much yet, but I have a new message dialog too.

• version 25

- Thumbnails should load just a bit quicker.
- Made some dict + set code neater
- A new end-of-filtering dialog makes the choices a bit more obvious
- When you exit from fullscreen/filter view, the main thumbnail view it was based on should select (and scroll to!) where you were. But filtering is less reliable, since you may have just deleted the last thing you were looking at!
- Better scrolling in thumbnail view
- Scrolling in thumbnail view will follow the current selection, so the focussed media will always be visible.
- Double right-clicking in filter now deletes instead of keeps (typo)
- Slightly less flickering when processing repo updates
- Made some form code neater
- A lot of the thumbnail gui code has been decoupled and pubsub'd. It is all neater behind the scenes, meaning I'll be able to extend it easier in future.
- I completely overhauled how my pubsub works. (pubsub is a messaging system inside the code). It is more flexible and easier for me to work with.
- system:mime was not working in v23, not sure why. It's fixed now.
- Some small speedup + neatening in db queries.
- You can now scroll to the end of a thumbnail page by pressing End. WARNING: If you have like 10,000 thumbs in the current pane, it is going to have to load all those.
- In the AutocompleteTagWrite object, which is the tag entry box in the manage tags dialog, there was a stupid bug that made many requests more expensive. It's fixed.
- Autocomplete now works off one character's input.
- I've swapped a bunch of expensive objects with easier staticbitmaps for now
- some A/C ordering changes

• version 24

- imageboard dumper prototype
- some db update stuff

version 23

- A typo was disallowing petition rescinds in certain cases.
- I've rearranged some of the RMB menu separators to reduce filter/archive misclicks.
- Tag pending upload has more accurate GUI response.
- You can now find similar images from RMB menu.
- Petition approvals are mirrored on an admin's client.
- The gui now accepts that admins can overrule deleted content.
- Couple tiny URL downloader gui bugs fixed

version 22

- system:similar_to
- Some backend namespace stuff
- Couple of gui presentation fixes
- I removed the clunky inbox and archive buttons, see how you feel about it.
- I tried to do ultimatelistctrl to have coloured tags (like a *booru), but it was too buggy and slow for the client's purposes, so I'll probably have to write my own control.
- system:mime exists, now. Not many mimes, yet, but have a play with it, see if you like it.

version 21

- On opening the URL downloader, the textbox input focuses.
- The URL downloader has some better update texts. It pauses nicely, etc…
- The URL downloader now orders urls in the order they are parsed. I forgot I was throwing the links in a set, losing all order!
- You won't see dupes in the URL downloaders' mediaresultswindow.
- Accounts should now correctly refresh on version update.
- When accounts fail to sync, it dumps the error in the log.

version 20

- The preview pane should be a little less schizophrenic.
- Some code cleaning up behind the scenes.
- Thumbnail download optimisation if they already exist locally.
- Icons will show/hide in fullscreen with mouse movement, just like 21/123.
- Weird archiving bug on import is fixed. It was display only.
- Manage tags dialog now works in fullscreen, lol.
- system:not_uploaded_to: now works
- services->review services now loads muuuuuch faster (after the first time)
- There was a typo in setting up initial server's accounts. Email me if you can't make new admins.
- Importing is more streamlined, with more shared code.
- Importing with tags dialog happens more naturally in the import process. You can now do it from drag and drop imports!
- I reordered the default collection dropdown. Your default may be changed, so check your options.
- You can now fetch an account by a deleted file (hash).
- Thumbnails' tags now obey tag precedence for deleted stuff. I hope.
- Modify Account Dialog now does its stuff.
- Collections thumbnails now show how many files are in the collection, just next to the icon.
- I've added to and edited the help extensively, replacing screenshots and updating the text.

version 19

- drop librarium nomenclature
- More intelligent prefetch in fullscreen - things should load muuuuuch faster
- A little more intelligent sum() calcs in autocomplete - non-namespaced tags now sum all their namespaced versions as well
- More rigorous client db transactions, just to make things more pretty and reliable.
- Moved db files to /db, logs to /logs
- Made inroads to getting review services to load faster, but version 20 will have more here.
- I've updated the frame view so the file info is draw on top of the image. It's a hack for now, inside the image-frame, but it is the only solution for wx's annoying 'no overlapping' policy.

version 18

- Some misc gui image info bugs fixed
- Review services dialog is less pubsub buggy
- Fullscreen image prefetch does more and hopefully more intelligently
- Entering a blank tag to quit tag dialog was broke when I updated autocomplete textctrls. I've fixed it.
- Petition page is fixed, so I can finally clear out the tag petitions!
- F10 was broke when the lib had deleted files lol! Fixed it.
- Review services is more responsive to repo permissions updates.
- Flash files are now correctly put in the inbox on import.
- Synchronise Repositories daemon no longer takes ten gui-blocking seconds to process every update! This was an annoying slowdown, but I figured out what it was and reduced it to <1s with a quick patch. The problem will return with private tag repos, but I'll deal with that when I come to it.
- Synchronise Accounts daemon has some better pubsub so it fires more intelligently. This in turn affects several other daemons, which should all be more responsive; e.g. when your permissions change, your pending thumbs/files should start downloading.
- Improved delete orphans logic for the fiftieth time! It is great now, and won't ever bug you in the gui; it runs invisibly on shutdown and usually only takes a few secs to do its job.
- Some of the gui updates triggered by the daemons are optimised so they block the gui less. Multiple stuff is bundled into single updates, those updates are less buggy, and the gui catches them better. Resetting the public tag repo will no longer lock up a slow system.
- A few versions ago I changed the way repos were registered with the gui, and I've fixed some of the resultant daemon-related errors. This mostly affects admins, which mostly means me!
- Download Thumbnails daemon was getting pretty slow for large dbs (100k+ files) because of an inefficient db query. This has been cleared up.
- I banged my head against the wall for a couple of hours getting nested notebooks to work, and I am now a better person for it. Review and Manage Services dialogs feature better panel management as a result. Their panels are also much richer, reporting loads more data about the respective service. You can now see how many mappings the public tag repo has without having to delve into sqlite db browser.
- I have unified panel and frame background colouring away from wx's bonkers idea of what things should be coloured. It should mean that everything is properly grey, but please tell me if your gui turns all black or transparent or whatever.
- 21457 is now 21,457 in pretty much all cases where it is appropriate.

- Fixed a bug with wx gauges not liking large ranges because of signed short or whatever underlying var the C code was using to store it. (That should have been done at the wx-python level, but whatever.)
- 'localhost' resolves to 127.0.0.1 when editing a service, mainly because windows freaks out at the former, deciding to take a couple seconds longer to make any raw connection.
- Dialogs and frames have better, multi-monitor-friendly positioning.
- Flash files are loaded from a temp folder beneath the install dir, rather than lib's server, for various reasons. This temp dir also manages file copy/paste operations. I may move this back to the OS's approved temp folder, but we'll see.

version 17

- Tag negation "-tag" works again
- You can now cancel a filter on the first image
- Some weird filenames that were breaking the new import page no longer do
- Autocomplete (either when searching for tags or adding new ones in) will now show the number of files tagged with that tag after the tag e.g. here
- Manage tag dialog now pops up slightly to the upper left, rather than the center (default) of the screen, so you can see what you are tagging. I may make this more intelligent in future, so it remembers where you last left it.
- I changed a couple small things in help regarding the new import.

version 16

- Numtags system predicate now takes into account which of current/pending tags buttons you have active.
- Thumbnail downloads fire more reliably.
- Downloading a file was causing funky thumbnail draw, which seems to be fixed.
- Delete Orphans daemon only fires on application close now, so post-filter db lag is gone.
- Daemons now use events, not conditions! lololol, I'm an idiot who can't scroll down a library doc. For the longest time I was like "I wish 'threading' had events!" and it does.
- Some account checks were missing in tag upload, but are now there.
- A gigantic db rewrite that altered pretty much every table and query.
- Calculation of tags for current selection now support collections, and is fast.
- Was deleting a repo broke in 15? If it was, sorry! Didn't catch it, but it's fixed with the new system now.

- Pubsub can do more, and is a little less 'argh, I think it works like that', and more OOP
- The way tags are held by a thumb can support multiple tag servers (but only does public tag repo for now)
- Repos are now a subclass of 'service'. I'll do more on this in the next version, including some nomenclature changes in the gui.
- Imports are now asynchronous! There's a new import page that'll open on any import. It's a little jittery, but it all works. If you import 3000 tiny files, your gui may lock up for a few secs since it'll be able to import faster than it updates the gui.

version 15

- manage tags dialog focus bug hotfix

version 14

- Selection's tags box now shows collections' singletons' tags
- Autocomplete now sources from the current thumbnails whenever it can (faster than db), and does so intelligently
- Couple of small display bugs fixed
- Some delete logic approved
- Repos do some more efficient inserts on POST content
- Petition optimisation, for auto-soft-approval before update gen
- Dialog parents, dialog parents everywhere
- DB cleanup of a couple of old methods
- Autocomplete Write (when adding tags in manage tags dialog) now sources from the entire network's tags (not just for the files on your hard drive) when you type enough letters to stump it. i.e. If you have no files tagged 'evangelion', but you do have a file tagged 'eve', it'll show 'eve' if you type 'ev', but if you type 'eva', it'll requery the db more widely, and give you 'evangelion'.
- system:ratio for searching for wallpapers or whatever. Combine with system:width>x for extra utility!
- You can now hit F3 (manage tags) in fullscreen, whether browsing or filtering.

version 13

- a couple of cpu-intensive DAEMONS schedule themselves better. (less slowdown after a big filter)
- image rendering prefetch improved
- smoother image rendering to screen (especially when zooming), less flickering
- petition approval has smoother, less buggy workflow
- new hashes and tags are added to the db muuuuuuuuch more efficiently. most public tag repo updates happen in a second or two, rather than 30+
- some of the internal commands are streamlined together into a single class
- ctrl-s fixed
- clear pending file repos pubsub fixed
- ctrl-r now shows/hides splitters
- thumbnail delete logic is better
- select logic is better
- deleted_remote_files bug fixed
- headache-inducing weakref repositories __eq__ bug in pubsub fixed
- 'with' keyword all over the place now. turns out wx has a bunch of objects with support it (including dialogs!)
- gridsizers are now all flexgridsizers
- the list of tags on the current selection now has a weight attached like 'flower (23)' to show how many of the current selection have that tag. it doesn't support collections' weight yet.
- revised some ambiguous repo error messages
- you can now have a main window with no pages open
- right click menus on a selection are vastly improved (lots more info), but still a way to go.
- on the import files dialog, you can now specify to auto-archive rather than send to inbox
- thumbnails now know where they are uploaded/pending/petitioned to and will show that info with different icons
- escape key now deselects
- manage repos dialog is completely revised. comments appreciated.

version 12

- You can now tell lib to search 'current tags' only, 'pending tags' only, or both.
- Tag search is more efficient.
- Search in general is more efficient.
- Search code is more beautiful and decoupled.
- Tags are stored in memory more beautifully, and a similar class holds repositories for thumbs also.
- Age predicate is fixed (twice!)

- Server code is more beautiful. DB-Server interface is decoupled into a mixin class to make it easier to understand and manage.
- Icons for pending and petitioned files, and for 'currently downloading'.
- Prefetch fixed, so nearby fullscreen images load just in case you want to see them.
- Fixed a redundant pending tags bug
- Fixed a swf tag petitions bug
- On boot, temp folder is cleared.
- The 'current tags' window is now union on the current selection, not intersect. If nothing is selected, it shows all the tags of the current selection. In a later version, I'll order this list by count-weight.
- All (I think) tag updates are synchronous-ish with the gui. As updates come in from the public repo, thumbs will update themselves appropriately. Same goes for repositories.
- Gui-code tidy-up
- On deleting a local file, preview clears.

version 11

- swf support, lol
- You can now backup repositories from the admin menu. Info in help.
- Under the help menu, a quick way to set up the beta repositories' credentials.
- Account sync was full of bugs, and is now less so.
- When you go fullscreen, the preview window should go blank, which should save a little cpu time when you are looking at gifs and your ears when you are looking at swfs.
- Auto-complete is muuuuuch faster than it was. Also more correct. Still more work to do, though.
- Librarium hosts an http server on 45874, which supports GET /file and /thumbnail just like a file repo. Utterly prototype, but you can test it right in your browser if you like. Go http://127.0.0.1:45874/file?hash=whatever when lib is running.
- Ctrl-T opens a new local page/tab, Ctrl-W closes the currently active
- Image display in preview and fullscreen has been revamped, and is now both a little buggy and much better.
- Bandwidth tracking both server- and client-side should be more reliable.
- Delete lag (the slowdown after you boot/delete something) should be reduced.
- Query lag is significantly reduced, and new operators for predicates have been added. You can now go ≈ on a bunch of values like size and width and duration to get 'approximately equal to', which actually means plus or minus about five percent.
- A stupid error on file upload is fixed. If you've been having problems seeing the files you uploaded in preview, try resetting the respective file repository.

- A few daemon bug fixes.

version 10

  - hotfix for tag updates

version 9

  - Pressing a cursor key when no thumb is focussed no longer throws an exception.
  - Hitting enter on a blank input in the manage tags dialog now quickly oks the dialog.
  - Annoying splitter exception on import is gone. It was a one-line fix, because I was being stupid.
  - All searches should be a little faster. Mostly effects searches with 1000+ results.
  - Modal dialogs (in particular) should be just a little more responsive now. Less stuff is being dumped on the wx gui thread, and all db handling is more parallel.
  - Slow petition approval fixed, made almost instant.
  - Image rendering has been revamped. It should be less stuttery now, but there is still more work to do.
  - Some pending tag update logic was fixed.
  - If the focused thumb goes from 'not local' to 'local', its preview will load. (its on my to-do to do the opposite: clearing preview on delete in remote search.
  - DB daemons rewritten completely, so now a whole bunch of stuff flushes more efficiently, more reliably, and with less gui-blocking. Things happen smoother now.
  - Can now create repository database backups via librarium. Sending the command (via the admin menu) tells the repo to lock and copy its .db files to .db.backup. I'll update help to explain this more.
  - Can now copy files from within librarium to the clipboard! Just select from the right-click menu or press Ctrl-C, and your selection will be copied to clipboard. You can then paste to any directory, for quick export to anywhere. The export command remains for exporting thousands of files.

version 8

  - Fixed a missing transaction on repo db update.

- Export path can now be on other hard drives. You'll probably want to keep it on the same drive if you have lib on a usb stick, though.
- Fixed a really stupid '<' for '>' typo that was causing a ton of gui blocking and repository slowdown.
- File and thumbnail downloads are much snappier.
- One person had a very odd error while trying to open manage repos. I don't understand what actually caused the traceback to happen, but I think I fixed it nonetheless.
- inbox/archive buttons above the searchbox are now mutually exclusive.
- Fixed the version number in lib's about window.
- Fixed the preview window so it remembers its last size properly. Also reduced its min size so you can make it tiny.
- Fixed the problem people were having with admin rights. See this post if you were having problems.
- The RegisterRepoRequest that was filling people's logs on upload should be fixed now. Turns out I didn't understand how wx.CallAfter works during modal dialogs! I threw in a horrible fix, but it works. I'll revisit it.

version 7

- db changes are done
- fixed a bug with posting tag petitions
- when typing tags in, autocomplete now harvests from pending tags as well as current
- the regex dialog's tag entry is now autocomplete
- leading/following whitespace removal for tags, and multi-whitespace -> single
- a little server-side pubsub neatening up
- when importing via dialog, there is now a checkbox to ignore subdirectories
- Added a quick readme.txt in the base_dir for people who choose to extract rather than install. (Everyone should check out the help at least once)

version 3->6 (never released)

- after much feedback, completely rewrite db
- general improvements to gui

- # version 2

  - general fixes and improvements

- # version 1

  - first true metadata beta
  - shit actually works

- # versions 0.91 -> 0.99

  - move to python and wx, and an application- rather than browser-based model
  - building on concepts
  - scrapping p2p, turning to client-server

- # early versions of beta

  - made db, playing around with LAMP
  - an initial p2p framework started

Revision #1
Created 12 March 2021 17:13:58 by CuddleBear
Updated 12 March 2021 18:24:07 by CuddleBear