

A Summer of Discontent Around ChatGPT's Upload Deletion Function

Originally published 29 September 2026 at <https://www.unite.ai/a-summer-of-discontent-around-chatgpts-upload-deletion-function/>



On March 23rd this year, OpenAI [introduced the Library](#) to ChatGPT. Now, instead of evaporating like skittish Snapchat messages, everything you uploaded to a chat session would be permanently saved – and used to train new AI models, assuming you let that [default setting](#) stand.

PDFs, images, pasted text (which became transformed into text files automatically, and saved), and anything else that the upload facility permitted, would hang around permanently, unless you took the trouble to delete them.

As the feature evolved, the Library's trash management process adopted the 'email' paradigm, where the act of deletion would simply move the files into a holding area; and from there, the items would finally be deleted thirty days later.

From May 5th, for Plus and Pro web users, GPT-5.5 Instant was able to [glean context from previous chats](#), uploaded and/or generated files (and, optionally, Gmail), in order to improve response quality.

Nine days later, OpenAI [introduced paid tiers](#), with varying data storage limits depending on what you paid; and suddenly the whole model shifted from email to that of iCloud, OneDrive, or any other frictionless, one-click storage-as-a-service solution designed to extract permanent and [ever-rising rent](#).

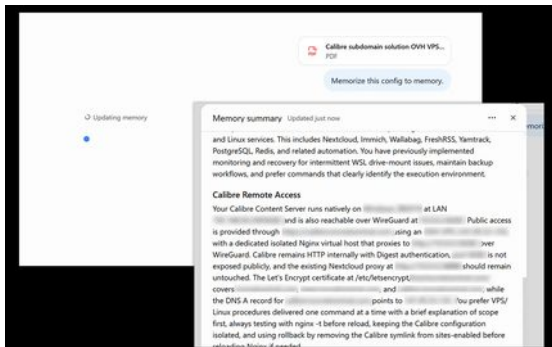


Time to cough up – hitting limits on OpenAI's ChatGPT product. [Source](#)

We Can Remember It For You, Retail

Here, however, OpenAI had an additional hook, since it became clear – certainly to me – that the new upload retention policy notably improved ChatGPT's continuity and recall of user-specific context (assuming that you don't routinely delete your uploads).

At the same time, ChatGPT's older long-term memory (LTM) interface became less directly editable, transformed into a [self-evolving but impermeable matrix of information](#) about you and your world. If you're lucky, and the wind is in the right direction, you can ask GPT to save a particular trait to LTM, and it will do it instead of saying that the LTM submit tool is not available in that particular chat:

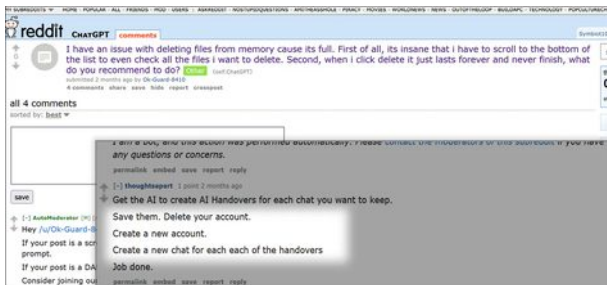


The new 'opaque' user-memory GUI, where individual facets cannot be discretely removed as before.

However, this older pool of data is no longer trivially editable; the locus of memory management seems instead to have shifted from LTM towards the accumulated documents, images, and other historical debris left behind by the user's previous chats.

These artifacts, many have noted in forums, are treated separately from the chats that spawned them; deleting a chat [does not delete](#) its associated images and files, and there is currently no option to make that happen.

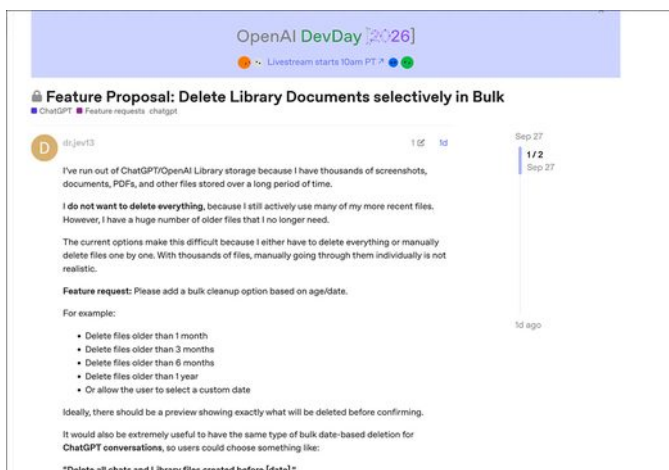
This could be interpreted as a 'dark pattern' inclining the user towards data retention, until they reach a point where the sheer volume of data is [so overwhelming](#), and the provided deletion workflows [so horrendously slow](#), that it is considered easier just to [delete](#) the ChatGPT account, and start over fresh.



The 'nuclear' solution seen increasingly where the friction around upload deletion is discussed. [Source](#)

So currently the trade-off presented to the user is that they must trust OpenAI with their files; take the trouble to manage and delete 'sensitive' uploads such as ID documents or client data (even though the current methods provided by OpenAI usually make it hard to navigate a high volume of uploaded files); and accept that if they do succeed in deleting their uploads, their future chats will lack the context that comes from that accumulated data.

Notwithstanding that most users are not data scientists, and may not easily be able to understand how to prune large file upload collections selectively and intelligently, the tools to address the issue seem inadequate to the task anyway:



At the time of writing, this latest request for better deletion management was only posted yesterday. [Source](#)

The upsell path, whether for OneDrive, iCloud or – now – ChatGPT, relies on threatening the user with actually having to do some curation of their endless data streams: the attachments that get auto-saved; the floods of photos that never get sifted and selected; the emails that might be needed for the accountant in 2038; and, now, the burgeoning and ballooning upload/generation cloud that makes AI actually work better; but which is retained on a self-serving company's terms, and within their infrastructure, rather than the users'.

Storage rentiers like OneDrive and iCloud rely on users preferring to pay more for increased storage quota than to go deep-sea diving through 10-15 years' worth of photos or other data, where the stakes can be high for errors and misjudgments.

Arguably, that's only a secondary revenue stream for ChatGPT's 2026 Library function, which instead is offering a workable solution to [LLM amnesia](#) in the form of a 'premium' service, devolved away from the old LTM facility. Though storage can be transferred to another paid provider in the rentier class, like Google Drive, there is currently no option to host your own data on WebDav, or other self-hosted solutions.

Struggling to Remember

One interesting question is *why* OpenAI is doing this. An obvious answer is that Google Gemini already features this mechanism in its paid subscriptions: as soon as you upgrade to a paid Gemini tier, your storage limits for the associated Google account increase notably, ready to hold all that sweet upload data hostage. From that standpoint, OpenAI is just leveling up a little.

However, even if OpenAI could realistically catch up to Google in terms of storage infrastructure, this tack seem very unlikely to save it from its [larger financial considerations](#), set to dwell in the trillions of dollars; arguably, [neither can advertising solve problems of this magnitude](#).

Therefore the remaining reasons could be *damage limitation* – where the additional revenue at least offsets *some* of the asymmetrical investment that OpenAI has had to undertake; *user-retention* – where the friction associated with migration becomes a deterrent to switching providers; or, less cynically, simply the provision of *something akin to long-term persistent memory* in human-AI relationships – which may be the central challenge of this third AI revolution; therefore, a pure value proposition.

A Word to the Wise

Many, including myself, are turning to the creation of 'handover' docs as an alternative to relying on unordered and uncured uploads to chat systems (uploads which may be infested with undesirable data points, such as network passwords):



The chat that informed this very disciplined and beautifully-ordered handover document was scrappy and not a little desperate, at times.

Using the [MD markdown format](#) (mainly because consumer AI frameworks seem to favor it, and it allows for a little more nuance and rationalization than a text document), one can at the end of an increasingly scrappy chat session ask the AI to summarize key points of the exchange into an overview that a later instance – or a different AI – could use to quickly assimilate the necessary knowledge and 'pick up' the task from that point.

One advantage of boiling down an exchange into a handover file of this kind is that most if not all of the uploads used in the chat can be summarized into the digest, and the originals safely deleted.

Another advantage is that one can selectively provide, update and delete these summary documents over time, retaining control of where the information is stored, and what data that information contains.

One would think that this kind of distillation was constantly going on behind the scenes of a frontier LLM provider; however, none of the major platforms' 'traditional' LTM systems seem to get any smarter or better-informed over time in quite the same way as when the user allows prodigious uploads to accumulate, and to provide rich – if messy and redundant – historical context.

Well, processing uploads into essential primers of the kind outlined above is not only resource-intensive, but could arguably be interpreted as some form of training, which many users have opted out of.

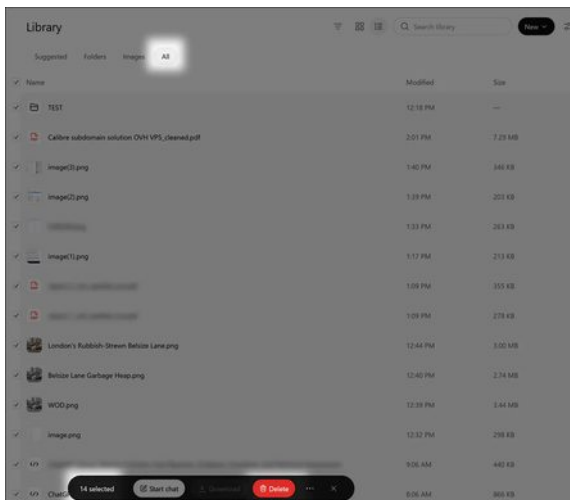
Conversely, uploaded images can be preserved with any [embeddings](#) extracted from them during a user chat, while documents are already text-based and suitable for reference. It's easier and cheaper to poll user metadata taken from these uploads at inference time than to reformulate the user's random upload mountain into a comprehensible database.

In the absence of a clearer rationale from OpenAI, this appears to be what is happening at the ChatGPT platform.

A Shyamalan Twist

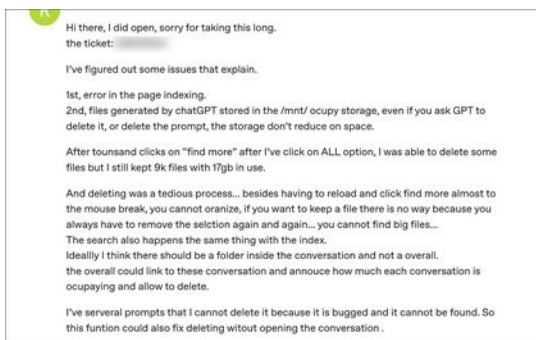
The 'summer of discontent' that this article refers to represents the frustration many users have felt about how hard it usually is to delete uploaded files from the Library in ChatGPT. To boot, terminology and related interface items seem to change nearly every day, quickly rendering [third-party](#) solutions such as a [Chrome web extension](#) to delete Library items, [non-functional](#).

One specific source of complaint is that the 'select' all button in the Library does not drill down and select all the items available – it only selects what has been lazy-loaded on the page:



When 'All' is not all – there may be thousands, or hundreds of thousands, of other files to deal with, only a thousand space-bar clicks away.

Therefore, if you have 90,000 uploaded items to delete, you are going to have to keep your finger on the space bar for a very long time – but not so long that the lazy loading loses context of the earliest items, or crashes the browser.

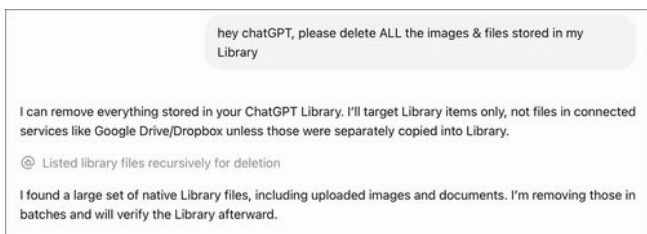


One user details the tedious process of selecting a large number of files in the library for deletion.

Yesterday, after a summer of gathering about 600 files into three specific projects (mostly related to home networking), I decided that my ChatGPT PDF exports – [not something that OpenAI makes at all easy](#) – and .MD summaries were sufficient for my long-term projects, versus the uncomfortable feeling of data about them accumulating in project folders in my ChatGPT account (and such project-associated files and images have been excluded from the usual routine upload deletions until a few days ago, in yet another GUI/usability reshuffle).

So I prepared myself for a tedious hour of button-pressing and oversight, and checked the forums to get up to date on user sentiment and tips around the issue.

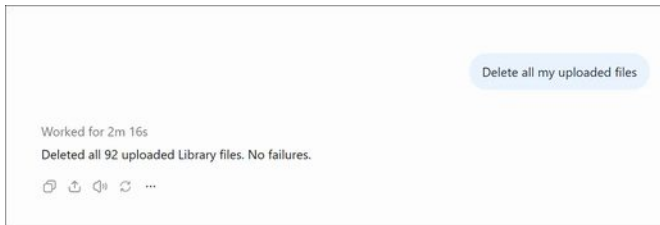
A [comment in one Reddit post](#) indicated that the user had just asked ChatGPT to delete the files itself, and it had complied:



ChatGPT solves the problem it faced users with all summer. [Source](#)

So I tried the same method on my own mere 600 uploaded files, and, after a certain amount of hedging and prevaricating, ChatGPT did indeed delete every single upload I had in my account, at one pass, and without requiring my attendance.

I tried it again this morning, on a test-set upload, and, again, it deleted all the files in one go, without the superintendence that many users had complained of as necessary:



In both cases I checked for [hallucinations](#), but the deletions were genuine. This means that there is GUI functionality in the GPT chat process itself that is (at least today) totally missing from the GUI itself.

It should be mentioned that my efforts to get ChatGPT to empty the trash all failed. Since the same aforementioned lazy-loading requirement is present for the trash, someone with 100k files will not easily be able to finalize their deletions. However, at least one need only wait a month, and it will happen.

It should also be considered that the interface and functionality of all the frontier LLMs mutate, shrink and expand on a daily basis, and that this functionality may not work for you; may disappear tomorrow; or may even evolve and improve.

First published Tuesday, September 29, 2026

Writer on machine learning, domain specialist in human image synthesis. Former head of research content at Metaphysic.ai, until its dissolution into DNEG's Brahma.ai.

Website: martinanderson.ai

Contact: martin@martinanderson.ai