

# Uses for Having Your GPT Save and Download Files

As previously mentioned, you can ask your GPT to save a file and provide a download link.

This is a pretty powerful capability for a few reasons. It's also one that we've only begun to explore.

Let's dig into it a bit deeper.

## **First, saving a file means you can create GPTs that transform files.**

For example, if you use Eleven Labs to create voice clones, you may want to trim an audio clip into multiple clips that are each only 4 minutes long because there is a 10 MB upload limit when you're training a custom voice.

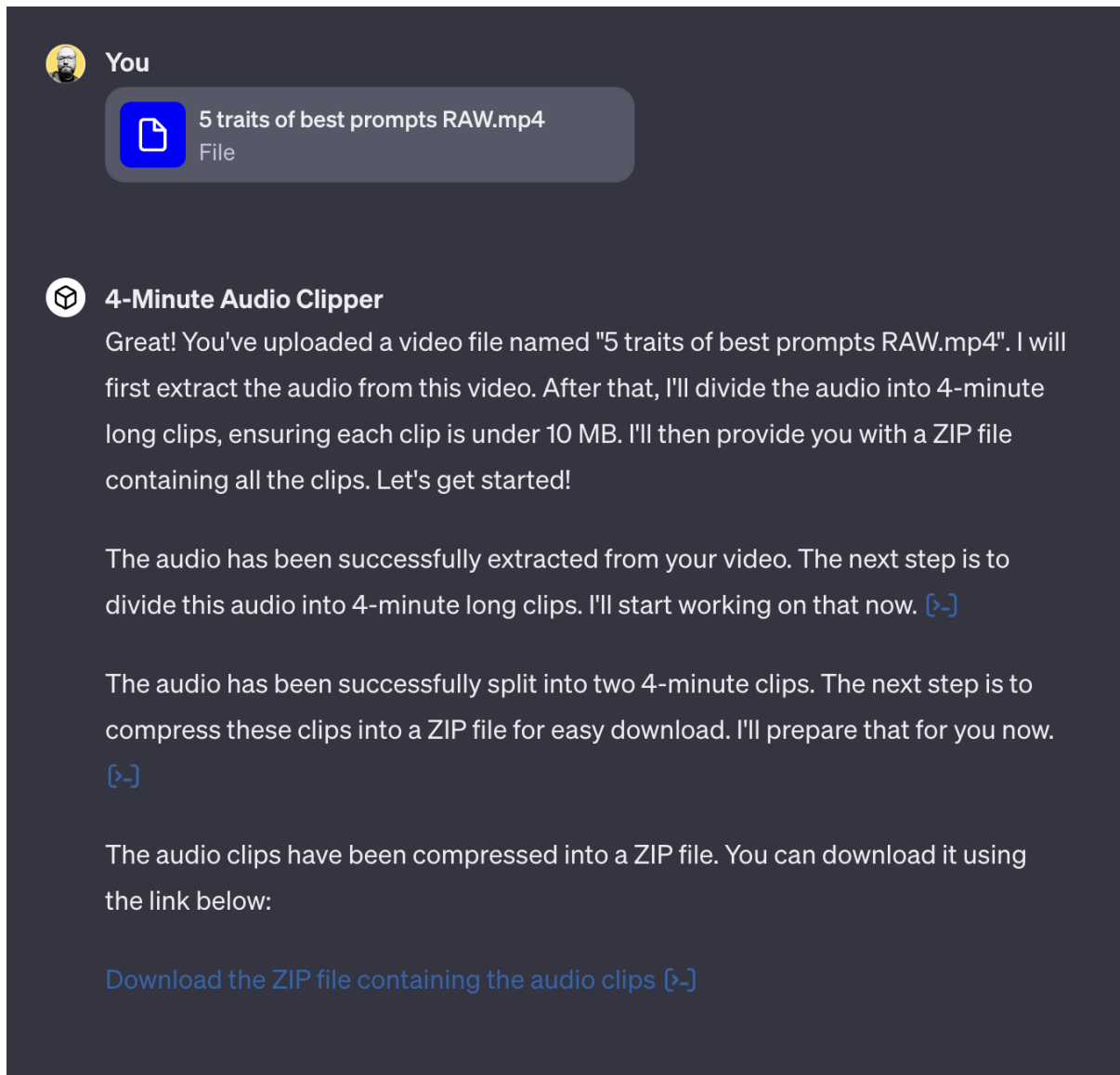
Well, you can program a GPT to do this and it should be able to save those files for you to download.

Here's what the instructions could look like:

```
For a given user input of a file, your goal is to break the audio down into a set of 4-minute long clips as MP3 files, each under 10 MB. You make them available for download to the user as a ZIP file.
```

```
If the input file is a video, you will have to extract the audio portion. If it is already audio, proceed as you would normally.
```

Don't forget to enable Code Interpreter as you'll need the GPT to execute some code to make this work.



The screenshot shows a chat window with a dark background. At the top, a user profile icon and the name "You" are visible. Below this, a file upload bubble contains a blue document icon, the filename "5 traits of best prompts RAW.mp4", and the word "File". The chat response begins with a cube icon and the title "4-Minute Audio Clipper". The text of the response is as follows:

Great! You've uploaded a video file named "5 traits of best prompts RAW.mp4". I will first extract the audio from this video. After that, I'll divide the audio into 4-minute long clips, ensuring each clip is under 10 MB. I'll then provide you with a ZIP file containing all the clips. Let's get started!

The audio has been successfully extracted from your video. The next step is to divide this audio into 4-minute long clips. I'll start working on that now. [>-]

The audio has been successfully split into two 4-minute clips. The next step is to compress these clips into a ZIP file for easy download. I'll prepare that for you now. [>-]

The audio clips have been compressed into a ZIP file. You can download it using the link below:

[Download the ZIP file containing the audio clips \[>-\]](#)

In theory you could use a similar process for resizing images, transforming text docs, and anything else you might need to do on a regular basis, provided there is a python library that the GPT can find that will do the task for you.

**Second, saving a file means you can store information for the GPT in a special place.**

The implications of this have only barely been explored.

But imagine you have a GPT that is going to have a long conversation with the user where it helps them edit some text they've written. The instructions could be to break the text into manageable sections, save them as files, work through them one-by-one, and to reassemble them for download when ready.

I haven't created or tested this specific approach, but you could in theory edit an entire book this way without having to copy and paste all the GPT outputs. You could even have it save a Changelog file with all the edits.

Here are two other interesting applications of this technique I can think of:

- Having the GPT generate multiple outputs and then be able to easily compare or synthesize them
- Having the GPT create detailed plans to accomplish a task and then iterate through that plan, checking off tasks as they are complete

It really depends on how complex you want to try to go.

## **Files Are \*Not\* Saved Between Sessions**

One important note here. Those files the GPT is creating are only stored for that specific chat session.

They're not being added to the base knowledge of the GPT.

So if you want a GPT to somehow retain information and evolve over time, you're going to have to download those files and upload them back in.

## **Files Have an 8,000 Token Limit**

As of the time of this writing, it appears that file retrieval is limited to 8,000 tokens (about 6,000 words). This is quite large and suitable for many cases, but anything longer than that will likely be truncated or only partially retrieved. This goes for files you upload into the knowledge as well as files you might be working with "in real time".

If you upload an entire book into the GPT, don't assume it will have the entire book in its context for every question you ask.

And if you use a GPT to write an entire book, similarly don't assume it will be referencing the whole thing every time.

## **Takeaway**

There are many creative and interesting applications I can think of for storing and retrieving files and making them available for download. This is definitely an area to experiment with.

Still, don't overcomplicate things. Sometimes a good 'ole regular prompt with its regular output is going to be best.