Guide to convert .ckpt models to .safetensors directly with Voldy (AUTOMATIC1111)'s UI

What are .safetensors in the first place?

You can read more about them in full detail here and further down the guide:
https://github.com/huggingface/safetensors

Basically, like their name implies, they're a much safer file format for distributing models.

Prerequisites

This guide assumes you already have a working install of the UI, there are better guides for that, like this one: https://rentry.org/voldy. This guide also assumes you're using Windows, if you're using Linux you probably already know what you're doing anyway.

If you already have a working install, but maybe haven't updated your UI in a while, getting .safetensors support is a simple as:

1. Open cmd or the Terminal in your main "stable-diffusion-webui" folder
2. Type `git pull`
3. Then type `pip install -r requirements.txt`
4. Wait for stuff to download and install

That's it, once everything is downloaded just launch the UI as usual. If you still get errors:

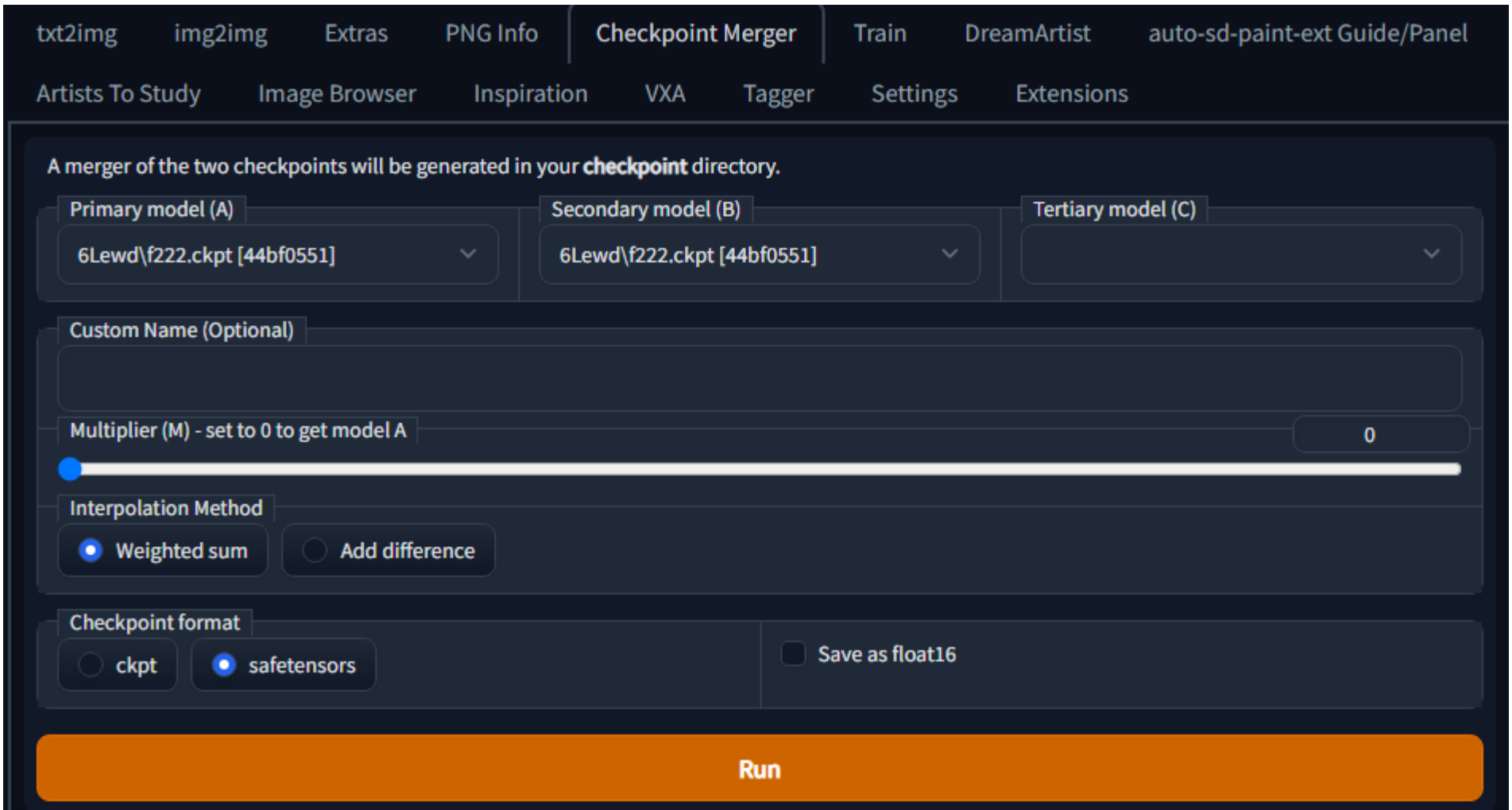
1. Open cmd again in your main "stable-diffusion-webui" folder
2. Type `venv\Scripts\activate`
3. Then type `pip install -r requirements.txt`

This will download/update the required dependencies inside the Virtual Environment (venv) which can fix issues.

How to convert your .ckpt model to .safetensors

1. Go to the "Checkpoint Merger" tab
2. Put the .ckpt model you want to convert in slot A
3. Put the same .ckpt model in slot B (technically it doesn't matter but just in case)
4. Put in a custom name (also doesn't matter, but note that if you leave it blank, the name will contain the name of both models used plus the difference, same as when merging any model, it's just a name anyway)
5. Put the "Multiplier" slider exactly at 0
6. Keep "Weighted Sum" selected, otherwise it'll error out because there's no C model selected
7. Finally in "Checkpoint format", select "safetensors"

Your Checkpoint Merger window should look like this



Then simply click Run and wait for the .safetensors model to be generated.

That's it, you can load the converted model and test to see if everything went right, then start using it normally as you would any other model. Note that you should still have the original .ckpt model, you can keep it as a backup or delete it later once you've confirmed everything works perfectly, up to you.

(Optionally, you can also convert to float 16 if you really want to, remember that FP16 can somewhat change outputs and it needs modern cards to run. It's not needed but the conversion should work.)

Can I merge .safetensors models with other .safetensors and .ckpt ?

Yes, you can merge both .safetensors with each other and even a .safetensors with a .ckpt, just be sure to save as .safetensors.

Why convert .ckpt to .safetensors?

No more fear of "pickles", AKA malicious Python code inserted into the models and no more need to scan for pickles. All current models contain pickles since they're a Python standard, but the word "pickle" became tied specifically to malicious code.

With .safetensors only the weights and specific data needed for generations are included. No additional unrelated and potentially malicious Python code can be included and run when loading .safetensors, like it can be done with .ckpt models.

Converting should be considered when distributing new models and merges going forward, as it avoids the minor paranoia when downloading a new model.

Is there a visual difference between .ckpt and .safetensors?

Nope, they output 100% the same images. Any difference in output is only caused by performance tweaks like --xformers, or also converting them to FP16 during the conversion process. Here's an example, one was generated with the .ckpt model, the other with the .safetensors model, management needs you to find the difference between them.

Do I need to convert all my older models?

Not really, only if you're (re)distributing them. If you're pointing someone to a download of Anything v3 for example, it's probably better to point them to the .safetensors version than the .ckpt version. If you're redistributing your own mix or new Dreambooth model, it's also better to share it as .safetensors.

Is there another advantage to .safetensors?

Supposedly faster model loading, though I personally haven't noticed much of a difference.

Is converting to .safetensors safe?

Unfortunately the conversion process still needs the .ckpt data to be loaded first, which means potentially loading pickles. Fortunately the UI already has had pickle checking for a while, and there are external pickle checkers.

https://github.com/zxix/stable-diffusion-pickle-scanner
https://github.com/lopho/pickle_inspector

Once models start being distributed only in .safetensors format, this won't be needed anymore. Which is why .safetensors should become the standard for model distribution. This does mean there will be a transition period, so it's still recommended to scan for pickles before converting a model if no .safetensors version is provided.

Are .safetensors themselves really safe?

Much safer than .ckpt at least. Do keep in mind that almost all file formats in history have been exploited in one way or another. The current advantage of .safetensors is that malicious arbitrary Python code can no longer be inserted directly and easily into the models, so another type of more advanced exploit would have to be found.

Can I mass convert?

Someone posted a script in the comments for the .safetensors pull request: https://github.com/AUTOMATIC1111/stable-diffusion-webui/pull/4930

Script:
https://gist.github.com/xrpgame/8f756f99b00b02697edcd5eec5202c59

Keep strongly in mind that converting outside of the UI means you don't get the included pickle protection the UI provides, so you need to have scanned everything you want to convert with an external pickle checker beforehand.

Can I train on .safetensors?

tbh this is the only thing I haven't tested yet, but hypernetworks and embeddings should work, maybe. Training in the Dreambooth extension probably doesn't work yet, you'd have to check their repo.

tl;dr?

Fuck pickles. Use .safetensors for distributing models, they're safe, make them common use.