ComfyUI Install Guide

Exhaustive ELI5 Setup Manual for Lunera Users

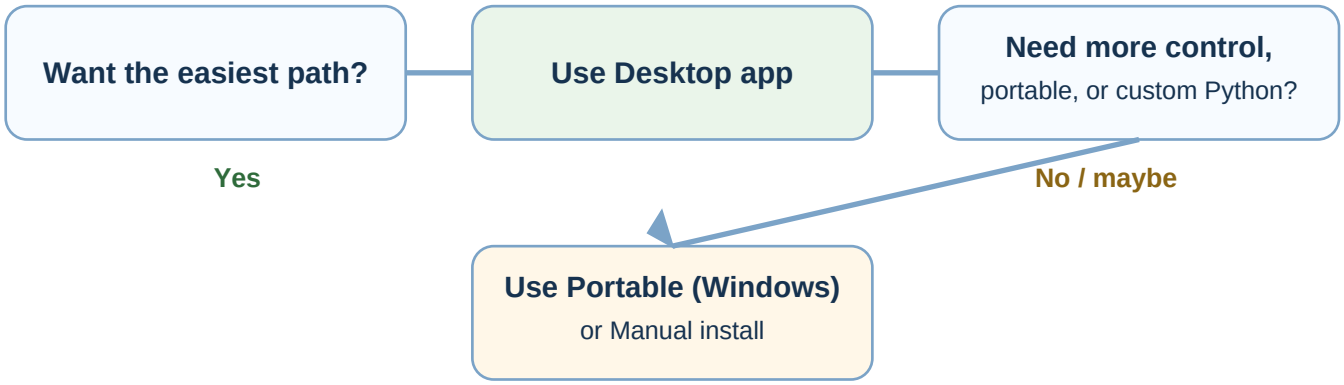
What this guide does

This guide shows the easiest install path first, then the deeper manual paths.
Every section explains what to click, what command to run, what success looks like, and what to do if it fails.

Good fit for this PDF

Use this if you want a downloadable support document for scenemethere.com, or if you want customers to get from zero to a working ComfyUI install with as little guesswork as possible.

Contents

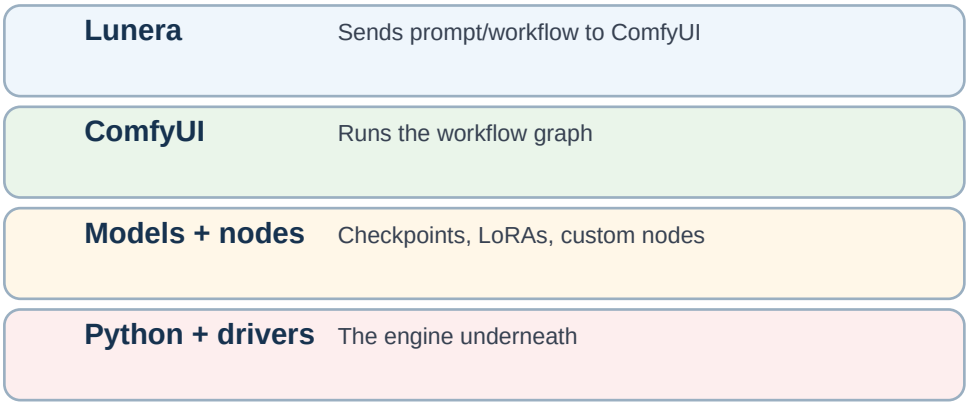


ELI5 translation: if you just want it to work with the least friction, use the Desktop app. Use Portable or Manual install only when you need more control, or when you specifically want to manage Python, folders, or launch options yourself.

1. Start here: choose the easiest install path

ComfyUI can be installed more than one way. The best choice depends on how much control you want.

Path	Best for	Why choose it	What you give up
Desktop app	Most people; simplest setup	It installs and manages the Python environment for you	Less “under the hood” visibility
Portable (Windows only)	People who want a folder-based install without a normal installer	Easy to move, easy to keep separate	Only for Windows
Manual install	People who want full control over Python, virtual envs, and launch flags	Maximum flexibility	Most steps; easiest to break



Think of this like a stack: each layer depends on the one below it.

Recommendation

For your website, tell beginners to start with the Desktop app first. Put the manual section lower on the page for advanced users. That reduces support requests.

2. Before you install anything

Do these checks before you click install. They prevent a lot of dead ends.

2.1 Know your machine

- On **Mac**, this guide assumes **Apple Silicon** (M1/M2/M3/M4).
- On **Windows**, the easiest and best-supported path is **NVIDIA GPU**.
- ComfyUI officially supports Windows, Linux, and macOS, and the Desktop app currently supports Windows and macOS (ARM/Apple Silicon).

How to verify this step succeeded

Mac: click Apple menu -> About This Mac. You should see an M-series chip.
Windows: open Task Manager -> Performance. Under GPU, you should see the NVIDIA model name if you have one.

2.2 Make sure you have enough space and patience

- Leave plenty of free disk space. Models are large.
- The install itself can be quick, but downloading models can take much longer than the app install.
- A “stuck” install is often just a large dependency download.

2.3 Get your basics in order

- For **Desktop app**: you mostly just need the installer.
- For **manual install**: install **Git** first, because ComfyUI's official manual install starts by cloning the repository.
- For some Windows manual setups, ComfyUI docs also note the Microsoft Visual C++ Redistributable if it is missing.

Plain-English warning

Do not mix three different install methods in the same folder. Pick one path, finish it, verify it works, and only then add models or custom nodes.

3. Easiest path: ComfyUI Desktop on Windows or Mac

This is the path to put first on a support page because it is the least technical. The official docs describe Desktop as a standalone install that automatically configures the Python environment and dependencies.

3.1 What Desktop means in normal words

- You download a normal app installer.
- You launch it like other software.
- On startup it prepares the needed Python pieces in the background.
- You do **not** have to manually build a Python environment first.

3.2 Desktop install steps

Step	What you do	What success looks like
1	Download the latest ComfyUI Desktop installer for your system.	The file finishes downloading with no corruption or unexplained browser problem.
2	Double-click the installer and go through the prompts.	A ComfyUI shortcut or app appears.
3	Launch ComfyUI Desktop.	You see initialization/setup instead of a crash.
4	Wait for first-run setup to finish.	The ComfyUI interface opens.

How to verify Desktop installed correctly

Open the app. If the interface opens and stays open, that is the first big win.
Then create or load a basic workflow. If the screen responds normally and you can see the graph area, the app itself is healthy.

3.3 If first launch feels slow

- That first launch may take longer than later launches.
- The app may still be downloading or unpacking dependencies.
- If the window opens and nothing crashes, give it a little time before assuming failure.

3.4 Common beginner mistake

People often think 'the app is broken' when really the app is installed but there is no model yet. An app can be installed correctly and still not generate images until a model is available.

4. Manual install on Mac (Apple Silicon)

Use this only if you want direct control. The official manual docs break the process into four parts: create a virtual environment, clone the repo, install dependencies, then start ComfyUI.

4.1 What a virtual environment is, ELI5

Think of a virtual environment as a private toolbox just for this app. It keeps ComfyUI's Python pieces from spilling into your whole computer.

4.2 Manual Mac checklist

- Install Git.
- Install Python or Miniconda.
- Create the environment.
- Clone the ComfyUI repository.
- Install dependencies.
- Run ComfyUI.

4.3 Commands

```
# 1) Open Terminal
# 2) Go to your home folder
cd ~

# 3) Clone ComfyUI
git clone https://github.com/comfyanonymous/ComfyUI.git
cd ComfyUI

# 4) Create a virtual environment (example using python3)
python3 -m venv venv

# 5) Activate it
source venv/bin/activate

# 6) Install ComfyUI requirements
pip install -r requirements.txt

# 7) Start ComfyUI
python main.py
```

How to verify each Mac step

Git clone worked: a ComfyUI folder now exists in your home folder.
venv creation worked: you now have a folder named venv inside ComfyUI.
activation worked: your Terminal prompt often shows (venv) at the front.
requirements install worked: pip finishes without a fatal error.
launch worked: Terminal shows the server starting, and you can open the local web address.

4.4 Mac gotcha to remember

A Mac manual install can be healthy even if generation is still slow at first. The first proof of life is that ComfyUI launches and the browser interface opens. Image speed comes later and depends on models and workflow complexity.

5. Manual install on Windows (NVIDIA GPU)

On Windows, beginners should usually use Desktop or Portable. Manual install is for people who intentionally want a Python environment and direct control over launch flags and packages.

5.1 Before you begin

- Install Git.
- Install Python.
- Open PowerShell or Command Prompt.
- Make sure you know which folder you want to keep ComfyUI in.

5.2 Commands

```
# 1) Open PowerShell
# 2) Go where you want ComfyUI to live
cd C:\

# 3) Clone ComfyUI
git clone https://github.com/comfyanonymous/ComfyUI.git
cd ComfyUI

# 4) Create a virtual environment
python -m venv venv

# 5) Activate it
venv\Scripts\activate

# 6) Install requirements
pip install -r requirements.txt

# 7) Start ComfyUI
python main.py
```

How to verify each Windows step

Clone worked: a C:\ComfyUI folder now exists.
venv worked: a venv folder exists inside C:\ComfyUI.
activation worked: your shell usually shows (venv).
launch worked: the console keeps running and ComfyUI opens in your browser or is reachable on localhost.

5.3 If Windows asks about security

A local app or script sometimes triggers a security prompt. Read it carefully. Do not bypass warnings blindly. Confirm you are using the official ComfyUI source or official ComfyUI Desktop download before you proceed.

5.4 Windows Portable option

ComfyUI's Windows Portable package is a self-contained folder-based version. In plain English: download it, extract it, then launch the included batch file for your hardware. It includes its own embedded Python, so it is simpler than a full manual install.

6. Your first launch and first image

This section is where people usually get lost, so keep it plain.

6.1 What you should see

- A ComfyUI window or browser page.
- A graph area with nodes.
- Menus or templates for loading a workflow.

6.2 Your first goal

Do not try to make the perfect image first. Your only job is to prove the machine can launch ComfyUI, load a workflow, and run one image.

6.3 First image checklist

Check	What to look for	What it means
Workflow loaded	Nodes appear in the graph	ComfyUI UI is functioning
Model selected	Checkpoint/model field is not blank	ComfyUI has something to generate with
Queue/Generate works	The queue starts and progress changes	The backend is responding
An image appears	A preview or saved output shows up	The full pipeline works

ELI5 tip

If the app opens but you cannot generate, that usually means 'missing model' rather than 'broken install.'

7. Models: where they go and how to know they loaded

A model is the heavy file that actually powers generation. No model, no image.

7.1 What folder matters most

Type	Typical purpose	Typical folder
Checkpoint model	Main image generation model	models/checkpoints
LoRA	Style or subject influence	models/loras
VAE	Image decoding helper in some workflows	models/vae
ControlNet / other add-ons	Workflow-specific helpers	Depends on node or model type

7.2 How to verify a model is actually seen

- Restart ComfyUI after adding files if the model list does not refresh automatically.
- Open the model dropdown in the node that expects that model.
- If the filename appears in the dropdown, ComfyUI sees it.
- If the filename does not appear, the file is either in the wrong folder, the wrong format, or ComfyUI needs a restart.

What success looks like
When you load a checkpoint node, the model you copied into the checkpoints folder appears in the dropdown list and can be selected without an error.

7.3 Biggest model mistake

Putting a file into the ComfyUI folder is not the same as putting it into the correct model subfolder. The path matters.

8. ComfyUI Manager and custom nodes

Custom nodes extend ComfyUI. Manager support is built into current ComfyUI, but for manual and portable installs the docs show that you need to install manager requirements and start with the `--enable-manager` flag.

8.1 Manual install example

```
# Activate the environment first
# Windows
venv\Scripts\activate

# macOS / Linux
source venv/bin/activate

# Install manager requirements
pip install -r manager_requirements.txt

# Start with manager enabled
python main.py --enable-manager
```

How to verify Manager works

Start ComfyUI with the manager flag. If the Manager area or related UI appears, or manager features are available, that step worked.

8.2 Installing custom nodes safely

- Only install custom nodes you trust.
- Add one major custom node set at a time, then relaunch and test.
- If ComfyUI breaks after a new node install, the newest node is the first suspect.

8.3 Simple test after any custom node install

Launch ComfyUI, load a workflow that uses the new node, and confirm the node name is recognized instead of showing missing-node errors.

9. Connect ComfyUI to Lunera

Once ComfyUI itself works, Lunera is the easy part.

Lunera setting	Value	How to verify
Local Rendering URL	http://127.0.0.1:8188	Lunera connection test passes or local render responds
Workflow JSON	Load the JSON that matches your ComfyUI workflow	Nodes line up and the workflow runs
Models / nodes	Must already exist in your ComfyUI install	Workflow does not fail on missing assets

ELI5 translation

Lunera is not replacing ComfyUI here. Lunera is talking to your already-running ComfyUI. If ComfyUI itself is broken, Lunera cannot magically fix it.

9.1 Best order of operations

- First prove ComfyUI works by itself.
- Then prove the workflow works inside ComfyUI.
- Only after that, load the same workflow into Lunera.

10. Verify everything with a final checklist

Question	Yes means	No means
Can I open ComfyUI?	The core app/server works	The install itself still needs attention
Can I load a basic workflow?	The UI is healthy	The launch may be incomplete or broken
Can I select a model from a dropdown?	ComfyUI sees the model file	The model is missing, misplaced, or unsupported
Can I queue one image?	Backend execution works	There is a workflow, model, or runtime problem
Can Lunera reach localhost:8188?	Local integration is alive	Connection or server issue

What 'fully working' really means

A fully working install is not just 'the app opened.' It means: app opens, model is visible, one workflow runs, one image is produced, and Lunera can connect if you are using it.

11. Fixing common problems

Problem	Most likely cause	First fix to try
App opens but no image generates	Missing model or broken workflow	Load a simpler workflow and confirm a checkpoint is selected
Model file copied but not visible	Wrong folder or restart needed	Move it to the correct subfolder and relaunch
ComfyUI worked yesterday but not today	A node, model, or update changed something	Revert the latest change or test with a basic default workflow
Lunera cannot connect	ComfyUI server is not running or wrong URL	Start ComfyUI first and confirm localhost:8188
Terminal still running	That is normal while ComfyUI is serving	Leave it running during use

11.1 The one-change rule

When debugging, change one thing at a time. Do not update Python, add five custom nodes, move folders, and swap workflows all at once. That makes the cause impossible to spot.

11.2 Your clean test

When everything feels messy, go back to a plain text-to-image workflow with one known-good checkpoint. If that works, the base install is fine and the problem is in the extra complexity.

12. Updating safely

ComfyUI updates often. Update with intention, not panic.

- Before a major update, back up your workflows and note which custom nodes you rely on.
- After an update, test one simple workflow first.
- If a custom node-heavy workflow breaks after updating, the custom nodes may need to catch up.

Safe update rhythm
1) Back up workflow JSON files.
2) Update ComfyUI.
3) Launch ComfyUI by itself.
4) Test a simple workflow.
5) Then test custom-node workflows.
6) Then test Lunera integration.

12.1 Why this matters

A lot of support pain comes from updating everything at once and then not knowing what changed. Keep your changes small and test after each one.

13. Source notes

This guide was written against official ComfyUI docs and official PyTorch install guidance current on March 25, 2026. Links below are included so you can refresh website copy later if ComfyUI changes.

Source	Why it matters
docs.comfy.org/installation/system_requirements	Supported operating systems and Desktop availability
docs.comfy.org/installation/desktop/windows	Desktop app behavior and setup
docs.comfy.org/installation/manual_install	Official manual-install sequence
docs.comfy.org/installation/comfyui_portable_windows	Portable package structure and launch files
docs.comfy.org/get_started/first_generation	First workflow and first image guidance
docs.comfy.org/manager/install	Manager requirements and enable flag
pytorch.org/install selector	Current PyTorch install command selection

For a public-facing support page, keep the website copy simple and point advanced users to a separate 'Manual Install' page.

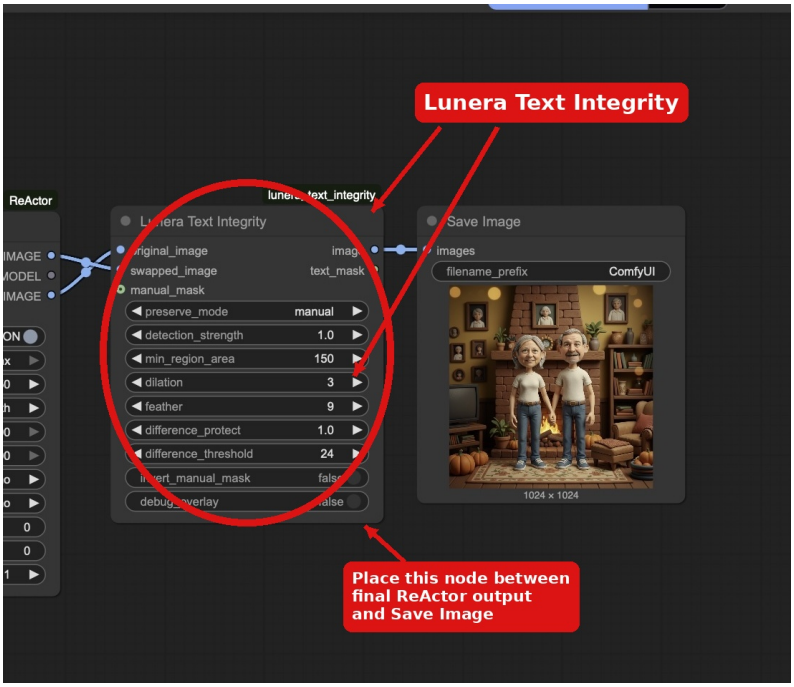
8.4 Lunera Text Integrity Node

This node is used when you want to protect readable text, signage, labels, or other lettering after a face-swap or image-to-image pass. In a Lunera workflow, the correct placement is after the last ReActor output and before the final Save Image node.

Why use it	It can help preserve text regions that would otherwise soften, warp, or break apart during the final stage of the workflow.

Where it goes in the graph

Place the **Lunera Text Integrity** node between the final **ReActor Fast Face Swap** node and **Save Image**. Feed the swapped image into this node, then route the node's image output into Save Image.



What the screenshot shows

The red circle and arrows mark the exact node in your workflow. That placement is important because it lets the node operate on the already-swapped image just before final output.

Good starting values

preserve_mode: manual
detection_strength: 1.0
min_region_area: 150
dilation: 3
feather: 9
difference_protect: 1.0
difference_threshold: 24

These settings are a practical baseline. Adjust only one control at a time and compare outputs. If you change several controls at once, it becomes hard to tell which setting actually improved or harmed text clarity.

8.5 Installing the Lunera Text Integrity Node

After you download the ZIP, extract it. The extracted folder needs to be copied into ComfyUI's **custom_nodes** directory so ComfyUI can load the node at startup.

1	Download the ZIP from the link below.
2	Unzip the archive so you get the extracted node folder.
3	Copy that extracted folder into your ComfyUI **custom_nodes** folder.
4	Restart ComfyUI completely. Do not just reload the browser tab.
5	Open your workflow. The Lunera Text Integrity node should now appear as a recognized node instead of a missing node.

Typical folder locations

macOS manual install	`~/ComfyUI/custom_nodes/`
macOS alternative manual path	`~/AI/ComfyUI/custom_nodes/`
Windows manual install	`C:\ComfyUI\custom_nodes\`
Windows portable install	`[Portable ComfyUI folder]\ComfyUI\custom_nodes\`

How to verify it loaded correctly

After restart, load the workflow again. If ComfyUI opens the graph without a missing-node warning and you can see **Lunera Text Integrity** in the chain before **Save Image**, the install worked. If the node is still missing, the folder is usually in the wrong location, nested one folder too deep, or ComfyUI was not fully restarted.

Download link

https://github.com/doww301/Lunera/releases/download/%23%24/lunera_text_integrity.zip

For most users, the cleanest method is: download ZIP → extract folder → move extracted folder into **custom_nodes** → restart ComfyUI → reopen the workflow and verify the node is present.