

Remix Workflow - Complete Guide

A SmartGallery DAM feature for generating directly from your gallery via the ComfyUI API

1. What is Remix Workflow?

Remix Workflow is a built-in feature of **SmartGallery DAM** that lets you take any image or video sitting in your gallery, read the **ComfyUI workflow embedded in its metadata**, tweak whatever you want (prompts, seeds, parameters, LoRAs, input images), and send a new generation job straight to your **ComfyUI server's API**. All of this happens without ever leaving the gallery or opening the full ComfyUI node graph interface.

In short, Remix turns your media library into a control panel for ComfyUI:

- **Extract** the workflow (API and/or UI format) baked into a PNG, video companion file, or a saved template.
- **Edit** it through three different levels of control (Auto-Form, My Panel, Nodepad).
- **Send it** back to ComfyUI to Queue a new generation, **Copy** it to your clipboard, or **Download** it as a `.json` file.

You can open Remix in three ways:

1. Press the **B** shortcut on any media file in the gallery that contains an embedded workflow.
2. Choose a saved template from the **Workflow Library**.
3. Open it from the **Tools** menu for a workflow-agnostic session (useful for browsing your Library).

2. The Remix Window at a Glance

When you open Remix, you'll see:

- **ComfyUI Server URL**, the address Remix will talk to (defaults to `http://127.0.0.1:8188`, editable and remembered locally per browser).
- **View toggle bar**, to switch between the three editing modes:  **Auto-Form**,  **My Panel**, and `{ }` **Nodepad**.
- **Footer actions**:  **Download**,  **Copy**,  **Save to Library**, and (when API data is available)  **Queue**.
- **Generation controls**: a random seed toggle and a generation counter, for firing off multiple queued jobs in one go.

If the file has **no embedded workflow**, or you're opening Remix from the Tools menu with nothing selected, you'll land on a helper screen pointing you to the Workflow Library or reminding you to use the **(B)** shortcut on a file that actually contains a workflow.

3. The Three Editing Views

3.1 📄 Auto-Form (the default view)

This is the "engine room." As soon as Remix loads a workflow, it automatically scans it and surfaces every editable field it recognizes, grouped into categories:

Category	What you can edit
💬 Prompts & Texts	Any text/string widget (positive/negative prompts, captions, etc.), with a "🔍 Expand" button that opens a full-screen editor. This is handy for long prompts or JSON-style prompt blocks (e.g. Ideogram), which are automatically pretty-printed and re-minified on save.
🎲 Seeds	Seed fields with a one-click "🎲 Random" button next to each.
⚙️ Parameters	Numbers, floats, and booleans (steps, CFG, denoise, switches, etc.) laid out in a compact grid.
🖼️ Input Images	Any input image referenced by the workflow, with a live thumbnail preview and a file picker to swap it out.

Each field can be **pinned** (📌 → ⭐) to add it to **My Panel**, your own custom dashboard (see 3.2).

3.2 🛠️ My Panel (Custom View)

My Panel only appears once you've pinned at least one field or favorited at least one node. It's your personal, decluttered dashboard:

- Pinned fields are pulled out of Auto-Form and displayed here, in whatever order you like.
- Click ⬆️ **Reorder** to drag-and-drop (or use the ▲/▼ buttons) and arrange fields exactly how you want them. This order is saved with your template.
- **Favorite nodes** (marked from the Nodepad, see below) show up here too, each with an "🔧 Edit in Nodepad" shortcut that jumps you straight to that node's raw JSON.

This is the view you'll want your future self (or teammates) to see when reopening a saved template: no clutter, just the knobs that actually matter for that specific workflow.

3.3 { } Nodepad (Raw Node Editor)

For everything Auto-Form can't (or shouldn't) expose automatically, there's the Nodepad, a full node-by-node JSON editor. See section 4 below for the full breakdown.

4. The Nodepad in Detail

The Nodepad is Remix's power-user mode: a two-pane, VS Code-style editor for the raw workflow JSON, node by node.

4.1 Layout

- **Left sidebar, Node list:** every node in the workflow, searchable by ID, title, class type, or by full-text search inside the node's JSON content (useful when you know a value but not which node holds it). Sort by **Flow Sequence** (the actual execution order, computed via a topological sort of the graph's dependencies) or by **ID / Favorites** (favorited nodes float to the top).
- **Left sidebar, Node Dictionary:** a live, interactive reference for the node currently selected. It fetches ComfyUI's `object_info` and shows every required/optional input, its type, allowed values (as a searchable dropdown you can click to inject directly), and any `min`/`max`/`step`/`default` constraints. If a value stored in the workflow (e.g. a checkpoint or LoRA filename) doesn't match anything currently installed on the server, it's flagged in red as **Missing Local File**.
- **Right pane, JSON editor:** a plain-text editor for the selected node's JSON (`inputs`/`widgets_values` depending on format), with live validation, line/column tracking, and an in-editor search bar.

4.2 Toolbar Tools

Button	What it does
 Revert Node	Discards your edits to the currently open node only.
 Revert All	Discards edits across every node in the workflow (with a confirmation prompt).
 Format	Pretty-prints the current node's JSON.
 Magic Fix	Attempts to auto-repair common JSON typos (trailing commas, unquoted keys) and reformats. Great for when you've been hand-editing and broke the syntax.

Button	What it does
 To ComfyUI	Converts the <i>entire</i> workflow from API format to a full ComfyUI UI-format graph (auto-layout, reconstructed links, default values inserted for anything omitted) so you can copy/download it and drop it straight into the ComfyUI canvas to visually inspect it.
 LoRA Synergy™	Opens the LoRA injection assistant, see section 5.

4.3 Favoriting & Healing Nodes

- Click ☆ **Favorite** in the node header to add a custom label and pin that node into **My Panel** for quick future access.
- For `LoraLoader` nodes specifically, a  **Remove & Heal Graph** button appears: it deletes the node and automatically rewires the `model`/`clip` connections around it, so removing a LoRA never leaves your graph broken.

4.4 The Magic Media Injector

Whenever you select a node that references an image, video, or audio file (and isn't a latent/checkpoint/mask type node), a small panel appears above the JSON editor showing a live preview of the referenced media and a file picker. Drop in a new file and Remix automatically injects the correct filename into the JSON for you, no manual key-hunting required.

4.5 API to UI Format Conversion

Workflows can exist in two shapes:

- API format**, the flat, ID-keyed structure ComfyUI's `/prompt` endpoint consumes.
- UI format**, the full node-graph structure (positions, links, widget arrays) ComfyUI's canvas understands.

If only API format is available and you choose **Copy** or **Download**, Remix will ask whether you want it converted to UI format first (so you can drop the result directly into the ComfyUI canvas). The conversion:

- Fetches ComfyUI's node dictionary (`object_info`) to know each node's expected inputs/outputs and default values.
- Topologically sorts nodes for a clean auto-layout.
- Reconstructs every link between nodes from the API's array-style input references.
- Fills in any widget values the API format omitted (because they matched the default).

LoRA Synergy is Remix's assistant for attaching, replacing, or swapping LoRAs into a workflow without editing JSON by hand, including matching them intelligently to your base model and surfacing their trigger words.

5.1 Even Better Together: ComfyUI-Lora-Manager Integration

LoRA Synergy works fully standalone, but it reaches its full potential when you also have ComfyUI-Lora-Manager installed on your ComfyUI instance.

When Lora Manager is detected, LoRA Synergy reads its local database directly and unlocks a much richer experience:

- **Instant thumbnails** for every LoRA, pulled straight from Lora Manager's cache instead of being fetched on demand.
- A direct, one-click link to the **CivitAI page** for each LoRA.
- **Smarter trigger word management**: official trigger words are read straight from Lora Manager's synced metadata, and combined with SmartGallery's own history of trigger words you've actually used, so you always get the most complete and most accurate list.

If Lora Manager is **not** installed, LoRA Synergy still works fully: matchmaking, chaining, and replacing LoRAs all function normally. You just get a reduced feature set for previews and trigger words (fewer thumbnails, no CivitAI shortcut, and trigger word suggestions based only on SmartGallery's own history rather than Lora Manager's richer synced database).

In short: no Lora Manager, LoRA Synergy still works great. With Lora Manager installed, it works in perfect synergy and becomes noticeably richer.

5.2 Opening It

From the Nodepad toolbar, click  **LoRA Synergy™** (API format only). Remix scans the graph for valid injection targets:

- **Checkpoint / UNET / Diffusion / GGUF loader nodes**, detected via known class-type keywords *and* via a smart heuristic that scans every input value on every node for filenames ending in `.safetensors`, `.ckpt`, `.pt`, or `.gguf` (this catches custom/community loader nodes that don't use standard naming).
- **Existing `LoraLoader` nodes**, so you can chain another LoRA after one already in the graph, or replace it entirely.

5.3 Matchmaking

Once a target is selected, Remix calls its LoRA intelligence backend to sort every installed LoRA into buckets:

Bucket	Meaning
 Perfect Match	Confirmed compatible by reading the LoRA's <code>.safetensors</code> header metadata.
 Proven Match	You've successfully used this LoRA with this checkpoint before (tracked in your local history).
 Possible Match	Inferred from folder structure or filename conventions, worth trying, not guaranteed.
 Incompatible	Detected as a different base architecture (e.g. an SDXL LoRA on a Flux checkpoint), collapsed by default to keep the list clean.

An architecture badge (e.g. `Arch: FLUX`) is shown when detection succeeds.

5.4 Trigger Words & Previews

Selecting a LoRA fetches:

- Its **preview image** (instant if Lora Manager is installed, otherwise pulled from cache when available).
- A direct **CivitAI Info** link and a **LoRA Manager** link (pointed at your configured ComfyUI server).
-  **Official trigger words** and  **historical trigger words** you've used before with this LoRA, each shown as a clickable chip that copies it to your clipboard *and* saves it into the floating  **Memory Triggers** widget (a draggable, collapsible panel that remembers trigger words across your session so you can quickly paste them into any prompt).

5.5 Injecting

Set your **Strength (Model)** and **Strength (CLIP)** values, then choose:

-  **Add After (Chain)**, which inserts a new `LoraLoader` (or `LoraLoaderModelOnly` for UNET-only targets) right after your target node and automatically rewires every downstream connection to pass through it.
-  **Replace Existing**, only available when the target is already a `LoraLoader`, which swaps its LoRA/strength values in place instead of adding a new node.

After injection, Remix jumps you straight to the new/updated node in the Nodepad so you can double-check it, and reminds you to add the LoRA's trigger words to your prompt node.

5.6 Configuring Model Paths

By default, SmartGallery looks inside your ComfyUI `models/` folder automatically. If you run ComfyUI in Docker, on a network drive, or with a custom `extra_model_paths.yaml`, set

these environment variables before starting the SmartGallery server so LoRA Synergy can find your files:

Variable	Purpose
<code>BASE_MODELS_PATH</code>	Root of your entire models folder, if moved.
<code>LORAS_PATH</code>	Granular override for LoRAs specifically.
<code>CHECKPOINTS_PATH</code>	Granular override for checkpoints.
<code>UNET_PATH</code>	Granular override for FLUX/SD3-style UNETs.

6. Workflow Library

The Library ( button, top-right of Remix) is where you save and reuse **templates**:

-  **Save to Library** captures the current workflow structure, your pinned My Panel layout, favorited nodes, and, your choice, either the *current modified values* or the *original untouched values*.
- Templates can be **renamed**, **deleted**, and **searched/sorted** (by name or by date) from the Library window.
- Opening a template loads it straight into Remix, ready to generate. This is the fastest way to build a repeatable "app" out of a complex workflow (e.g. "Flux Portrait", "Wan2.1 Img2Vid") that you or your team can reuse without ever touching raw JSON.

7. Sending the Job: Queue, Copy, Download

At the bottom of the Remix window:

-  **Queue** (only shown when API-format data is available) sends the job directly to your ComfyUI server. You can set a **Generations** count to fire off several jobs in a row (with **Random seed** enabled: using a fixed seed with multiple generations will just re-queue duplicates, and Remix warns you about this).
 - Every queued job is monitored live via ComfyUI's WebSocket (`/ws`) connection, showing progress percentage, current executing node, queue position, and a final success/failure status.
 - If ComfyUI rejects the workflow as invalid, Remix offers a  **Try Autofix** button that attempts to repair broken/renamed values automatically, prompting you to choose a replacement when it can't decide on its own.
-  **Copy** and  **Download** work with either API or UI format, converting between them on the fly when needed (see 4.5).

- ★ **Save to Library** stores the current state as a reusable template (see section 6).
-

8. Troubleshooting

"Clipboard copy failed" or nothing gets copied. Remix tries the modern Clipboard API first, then falls back to a classic `execCommand('copy')` method that works even over plain HTTP (e.g. Docker deployments without HTTPS). If both fail, usually because too much time passed between your click and the copy attempt (e.g. during an API to UI conversion), Remix now pre-arms the copy at the exact moment you confirm the conversion, and as a last resort opens a manual "Click Here to Copy" window so you're never left without a way to grab your JSON.

"Missing Local File" warnings in the Node Dictionary. This means a checkpoint/LoRA filename stored in the workflow doesn't exist on the connected ComfyUI server. Check your server URL, make sure the file is actually installed, or see section 5.6 if you're using non-standard model paths.

"Dictionary load failed" / can't reach ComfyUI. LoRA Synergy and the Node Dictionary both require a live connection to your ComfyUI server to read its `object_info`. Double-check the **ComfyUI Server URL** field at the top of the Remix window and confirm ComfyUI is actually running and reachable from the machine running SmartGallery.

That's Remix Workflow, end to end: pull a workflow out of your gallery, shape it at whichever level of control you need (quick Auto-Form tweaks, a curated My Panel, or full Nodepad control with LoRA Synergy), and send it straight back to ComfyUI.