

Aura Visual Studio

dayshafreeman@auravisualstudio.com
www.auravisualstudio.com

Technical Guide: Local_AI_Branding_Node // v.04

Technical Blueprint: CRM & AI Node Synchronization

1. Objective: Infinite Asset Consistency

The goal is to move away from the "lottery" of cloud-based AI. By utilizing local hardware (RTX 4090/3090 class), you anchor your brand's visual identity through local models. This ensures every generated image—from marketing assets to digital personas—maintains identical structural features, color palettes, and lighting logic.

2. The Character Anchor: LoRA Training

Generic models drift. To achieve consistency, you must train a **LoRA (Low-Rank Adaptation)** on your specific brand assets or persona.

- **Dataset Prep:** 20–50 high-quality images. Use varied angles but consistent lighting.
- **Captions:** Use a unique trigger word (e.g., **AuraNodeV1**) to isolate the brand features from the base model's noise.
- **Execution:** Train locally using **Kohya_ss** or **OneTrainer**. This "weights" the AI to prioritize your specific brand geometry over all other possibilities.

3. Compositional Rigidity: ControlNet & IP-Adapter

Prompting alone is insufficient for professional workflows. You must use spatial constraints to dictate the "where" and "how" of the image.

2

- ControlNet (Canny/Depth): Use these to lock in specific layouts or poses from existing sketches or 3D renders. It prevents the AI from "reimagining" the composition.
- IP-Adapter (FaceID/Plus): Uses a reference image as a visual prompt rather than text. This maintains "identity glue" when the LoRA isn't enough to handle complex environment shifts.

4. The Prompt Architect: Local LLM Integration

Stop writing manual prompts. Use a local LLM (via **Ollama** or **LM Studio**) to act as a "Prompt Engineer" that understands your brand's technical constraints.

- **System Prompting:** Feed the LLM your brand's technical manual.
- **Function:** Input a simple concept (e.g., "AI agent in server room"). The LLM outputs a high-fidelity Stable Diffusion prompt formatted with your LoRA triggers, negative embeddings, and lighting technicals (e.g., "rim lighting, 8k, cinematic anamorphic").

5. High-Fidelity Output: Tiled Diffusion & Upscaling

Raw AI outputs are usually 1024x1024—too low for professional use.

The Workflow:

1. Generate base image at low res.
2. Use **Ultimate SD Upscale** or **Tiled Diffusion** to increase resolution to 4k.
3. Apply a low-denoising pass (0.3–0.4) to add texture and skin/surface detail without changing the base identity.

6. Operational Hardware Checklist

To run this node efficiently without "Out of Memory" (OOM) errors:

1. **VRAM:** Minimum 12GB (16GB+ recommended for XL models).

3

2. **Environment:** Stable Diffusion WebUI (Forge) or ComfyUI for complex node-based automation.
3. **Storage:** NVMe SSD for fast model switching (Checkpoints are 6GB+ each).
4. **Backend:** Linux (Ubuntu/Pop!_OS) preferred for 5–10% faster generation speeds over Windows.

Logic Summary

Do not ask the AI for a miracle. Give the AI a blueprint (LoRA), a skeleton (ControlNet), and a technical prompt (LLM), then upscale the result into a professional asset.