

Hermes + Local LLM

Hermes Agent setup with a Local LLM on an old Nvidia GTX 1070 Super 8GB VRAM card on a Hex-Core Intel CPU.

STARTED

01/10/2026

BY

Michael Pope and Danch

TAGS

hobby

PRINTED

13 Oct 2026

Hermes + Local LLM

Setup LM-Studio 0.4.24-2

Publisher	unsloth
Params	8B
LLM Model	qwen3-8b-128k-GGUF
Quant	Q4_K_S
Capabilities	Tool use, Reasoning
Domain	LLM
Size on disk	4.80GB
Context Length	65536
K Cache Quantization Type	Q4_0
V Cache Quantization Type	Q4_0

Setup Hermes TUI

Install Hermes through Omarchy Menu (SUPER+SPACE, Hermes, Enter)

Run setup = Y

```
It looks like Hermes isn't configured yet -- no API keys or providers found.
```

```
Run: hermes setup
```

```
Run setup now? [Y/n] 
```

image13.png 8 KB

Select 'Full Setup'

```
How would you like to set up Hermes?
```

```
↑↓ navigate  ENTER/SPACE select  ESC cancel
```

- ☒ Quick Setup (Nous Portal) – free OAuth login, no API keys, model + tools (recommended)
- ☐ **Full setup – configure every provider, tool & option yourself (bring your own keys)**
- ☐ Blank Slate – everything off except the bare minimum; opt in to each capability

image12.png 17.6 KB

Select LM-Studio

```

Select provider:
↑↓ navigate  ENTER/SPACE select  ESC cancel

(●) Nous Portal (Everything your agent needs, 300+ models with bundled tool use)
(○) Fireworks AI (OpenAI-compatible direct model API)
(○) OpenRouter (Pay-per-use API aggregator)
(○) Mixture of Agents (named presets; aggregator acts after reference models)
(○) NovitaAI (Cloud: Model API, Agent Sandbox, GPU Cloud)
→ (○) LM Studio (Local desktop app with built-in model server)
(○) Anthropic (Claude models via API key or Claude Code)
(○) OpenAI ▸ (Codex CLI or direct OpenAI API)
(○) Qwen ▸ (Qwen Cloud / DashScope, Coding Plan & Qwen CLI OAuth)
(○) xAI Grok ▸ (Direct API or SuperGrok / Premium+ OAuth)
(○) Xiaomi MiMo (MiMo-V2.5 and V2 models: pro, omni, flash)
(○) Tencent TokenHub (Hy3 Preview via tokenhub.tencentmaas.com)
(○) NVIDIA NIM (Nemotron models via build.nvidia.com or local NIM)
(○) GitHub Copilot ▸ (GitHub token API or copilot --acp process)
(○) Hugging Face Inference Providers
(○) Google AI Studio (Native Gemini API)
(○) Google Vertex AI (Gemini via GCP; OAuth2 service account or ADC, GCP billing/quotas)
(○) DeepSeek (V3, R1, coder, direct API)
(○) Z.AI / GLM (Zhipu direct API)
(○) Kimi / Moonshot ▸ (Coding Plan, Moonshot global & China endpoints)
(○) StepFun Step Plan (Agent / coding models via Step Plan API)
(○) MiniMax ▸ (Global, OAuth Coding Plan & China endpoints)
(○) Ollama Cloud (Cloud-hosted open models, ollama.com)
(○) Arcee AI (Trinity models, direct API)
(○) GMI Cloud (Multi-model direct API)
(○) Kilo Code (Kilo Gateway API)
(○) OpenCode ▸ (Zen pay-as-you-go or Go subscription)
(○) AWS Bedrock (Claude, Nova, Llama, DeepSeek; IAM or API key)
(○) Azure Foundry (OpenAI-style or Anthropic-style endpoint, your Azure AI deployment)
(○) custom (direct API)
(○) DeepInfra – 100+ open models, pay-per-use
(○) Upstage (Solar API)
(○) Custom endpoint (enter URL manually)
(○) Configure auxiliary models...
(○) Leave unchanged

```

image1.png 129 KB

Hit enter on LM_API_KEY to accept default

```

Warning: No inference provider configured. Run 'hermes model' to choose a provider and model, or set an API key (OPENROUTER_API_KEY, OPENAI_API_KEY, etc.) in ~/.hermes/.env. Falling back to auto provider detection.

Current model: (not set)
Active provider: none

No LM Studio API key configured.
LM_API_KEY (Enter for no-auth default 'dummy-lm-api-key'): 

```

image16.png 23.1 KB

Hit enter on Base URL to accept default

```

Base URL [http://127.0.0.1:1234/v1]: 

```

image19.png 3.63 KB

Select default model as 'qwen3-8b-128k'

```
Select default model:
↑↓ navigate  ENTER/SPACE select  / search  ESC cancel

→ (●) qwen3-8b-128k
  (○) qwen/qwen3-4b-2507
  (○) Enter custom model name
  (○) Skip (keep current)
```

image7.png 11.6 KB

Select terminal backup 'keep current local'

```
Select terminal backend:
↑↓ navigate  ENTER/SPACE select  ESC cancel

  (○) Local - run directly on this machine (default)
  (○) Docker - isolated container with configurable resources
  (○) Modal - serverless cloud sandbox
  (○) SSH - run on a remote machine
  (○) Daytona - persistent cloud development environment
  (○) Singularity/Apptainer - HPC-friendly container
→ (●) Keep current (local)
```

image3.png 16.6 KB

Select 'Telegram' from the platforms

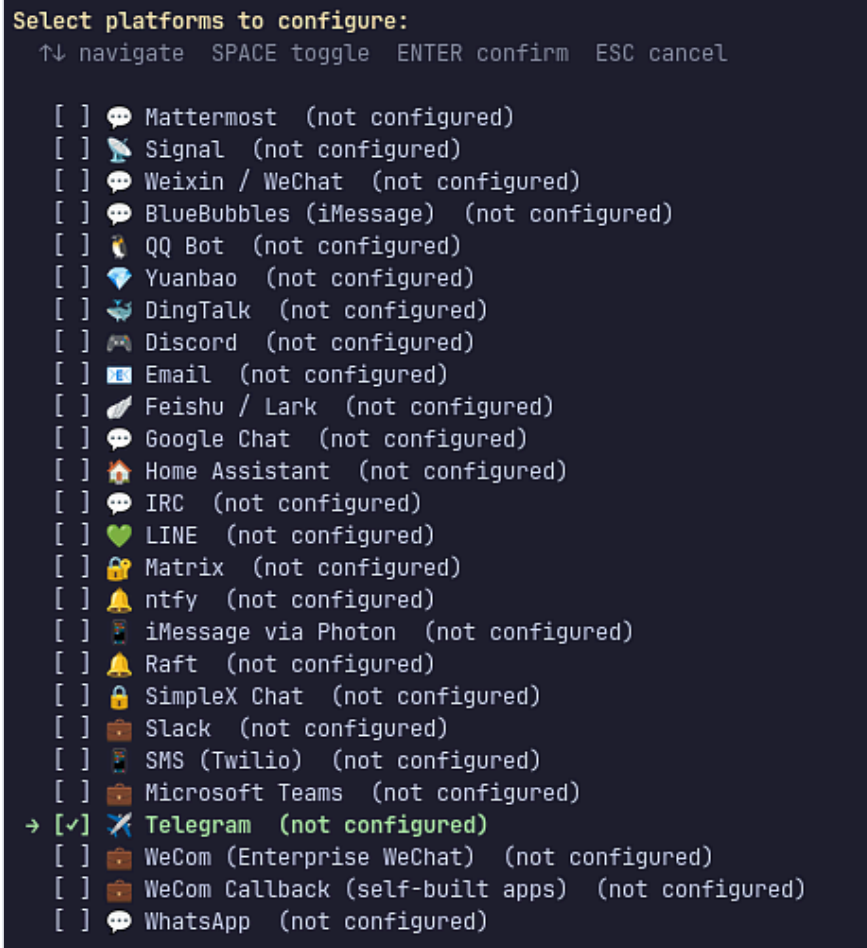


image15.png 56.9 KB

Select 'Automatic' for the Telegram bot

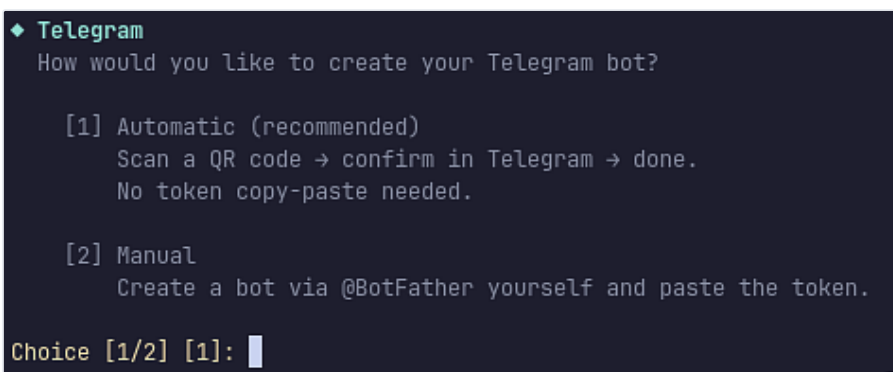


image11.png 13 KB

Open up the link on my mobile to link to telegram

```
Contacting Hermes Telegram onboarding service: https://setup.hermes-agent.nousresearch.com
✓ Pairing created
Rendering QR code...

Scan this QR code with your phone, or open the link below:

(Install 'qrcode' for a scannable QR code: pip install qrcode)

Link: https://t.me/NousHostedHermesBot?start=pair_71sr7wmlw3yvd6ns

When Telegram opens, tap 'Create Bot' to confirm.
(You can edit the bot display name before confirming)

⋮ Waiting for bot creation... (44s remaining) █
```

image2.png 26.1 KB

Get the API token in telegram

Go to BotFather in Telegram

Type '/mybots'

Select the hermes one

Select API Key

Copy the API manually back into hermes terminal

```
Telegram bot token: *****
✓ Telegram token saved

🔒 Security: Restrict who can use your bot
To find your Telegram user ID:
1. Message @userinfobot on Telegram
2. It will reply with your numeric ID (e.g., 123456789)

Allowed user IDs (comma-separated, leave empty for open access): █
```

image17.png 14.9 KB

Leave the Home channel ID empty for cron job results, etc

```
📁 Home Channel: where Hermes delivers cron job results,
cross-platform messages, and notifications.
For Telegram DMs, this is your user ID (same as above).
You can also set this later by typing /set-home in your Telegram chat.
Home channel ID (leave empty to set later): █
```

image20.png 10.4 KB

Setup systemd telegram = 'Y'

```
Install the gateway as a systemd service? (runs in background, starts on boot) [Y/n]:
```

image8.png 4.01 KB

Run gateway as a user service

```
Choose how the gateway should run in the background:
```

```
↑↓ navigate ENTER/SPACE select ESC cancel
```

```
→ (●) User service (no sudo; best for laptops/dev boxes; may need linger after logout)
  (○) Skip service install for now
```

image9.png 12.8 KB

Start the service now

Select the Tools, I added 'Home Assistant' to the defaults

```
Tools for CLI
```

```
↑↓ navigate SPACE toggle ENTER confirm ESC cancel
```

```
[✓] 🔍 Web Search & Scraping (web_search, web_extract)
[✓] 🌐 Browser Automation (navigate, click, type, scroll)
[✓] 💻 Terminal & Processes (terminal, process)
[✓] 📁 File Operations (read, write, patch, search)
[✓] ⚡ Code Execution (execute_code)
[✓] 👁 Vision / Image Analysis (vision_analyze)
[ ] 📺 Video Analysis (video_analyze (requires video-capable model))
[✓] 🎨 Image Generation (image_generate)
[ ] 📺 Video Generation (video_generate (text/image/reference))
[ ] 🐦 X (Twitter) Search (x_search (requires xAI OAuth or XAI_API_KEY))
[✓] 🗣 Text-to-Speech (text_to_speech)
[✓] 📖 Skills (list, view, manage)
[✓] 📅 Task Planning (todo)
[✓] 🧠 Memory (persistent memory across sessions)
[ ] ⚙ Context Engine (runtime tools from the active context engine)
[✓] 🔍 Session Search (search past conversations)
[✓] ? Clarifying Questions (clarify)
[✓] 👤 Task Delegation (delegate_task)
[✓] ⌚ Cron Jobs (create/list/update/pause/resume/run, with optional attached skills)
[✓] 🏠 Home Assistant (smart home device control) [no API key]
→ [ ] 🎵 Spotify (playback, search, playlists, library)
[ ] 🚗 Yuanbao (group info, member queries, DM)
[✓] 🖱 Computer Use (macOS/Windows/Linux) (background desktop control via cua-driver)
```

image10.png 67.1 KB

Choose a provider = Local Browser

```
Choose a provider:
↑↓ navigate ENTER/SPACE select ESC cancel

→ (●) Local Browser [★recommended · free] – Headless Chromium, no API key needed
  (○) Nous Subscription (Browser Use cloud) [subscription] – Managed Browser Use bi
  (○) Camofox [free · local] – Anti-detection browser (Firefox/Camoufox)
  (○) Browser Use [paid] – Cloud browser with remote execution
  (○) Browserbase [paid] – Cloud browser with stealth and proxies
  (○) Firecrawl [paid] – Cloud browser with remote execution
  (○) Skip – keep defaults / configure later
```

image5.png 18.4 KB

Went with defaults for Home assistant, just hit enter

Choose a provider (LLM) = skip

```
Choose a provider:
↑↓ navigate ENTER/SPACE select ESC cancel

(●) Nous Subscription [subscription] – Managed FAL image gen
(○) DeepInfra [paid] – FLUX, Qwen-Image, ... – live catalog fr
(○) FAL.ai [paid] – Pick from flux-2-klein, flux-2-pro, gpt-
(○) Krea [paid] – Krea 2 foundation model – Medium ($0.03),
(○) Nous Portal (image) [subscription] – Reference-grounded
(○) OpenAI [paid] – gpt-image-2 at low/medium/high quality t
(○) OpenAI (Codex auth) [free] – gpt-image-2 via ChatGPT/Cod
(○) OpenRouter (image) [paid] – Gemini Flash Image & more vi
(○) xAI Grok Imagine (image) [paid] – grok-imagine-image - t
→ (○) Skip – keep defaults / configure later
```

image18.png 25.1 KB

Select a TTS = Microsoft Edge

```
Choose a provider:
↑↓ navigate ENTER/SPACE select ESC cancel

→ (●) Microsoft Edge TTS [★recommended · free] – Good quality, no API key needed [active]
  (○) Nous Subscription [subscription] – Managed OpenAI TTS billed to your subscription ★via Nous Portal (login on select)
  (○) OpenAI TTS [paid] – High quality voices
  (○) xAI TTS – Grok voices – uses xAI Grok OAuth or XAI_API_KEY
  (○) ElevenLabs [paid] – Most natural voices
  (○) Mistral (Voxtral TTS) [paid] – Multilingual, native Opus
  (○) Google Gemini TTS [preview] – 30 prebuilt voices, controllable via prompts
  (○) KittenTTS [local · free] – Lightweight local ONNX TTS (~25MB), no API key
  (○) Piper [local · free] – Local neural TTS, 44 languages (voices ~20-90MB)
  (○) DeepInfra TTS [paid] – Chatterbox, Qwen3-TTS, ... – live catalog from api.deepinfra.com
  (○) Skip – keep defaults / configure later
```

image6.png 46.7 KB

Choose DuckDuckGo for search provider

```

Select Search Provider:
↑↓ navigate  ENTER/SPACE select  ESC cancel

(●) Nous Subscription [subscription] - Managed Firecrawl billed to your subscription ★via Nous Portal (login on select)
(O) Firecrawl Self-Hosted [free · self-hosted] - Run your own Firecrawl instance (Docker)
(O) Brave Search (Free) [free] - Free-tier API key - 2k queries/mo, search only.
→ (O) DuckDuckGo (ddgs) [free · no key · search only] - Search via the ddgs Python package - no API key (pair with any extract provider)
(O) Exa [paid] - Semantic + neural web search with content extraction.
(O) Firecrawl [paid · optional gateway] - Full search + extract; supports direct API and Nous tool-gateway routing.
(O) Parallel [paid] - Objective-tuned search + parallel page extraction.
(O) SearXNG [free · self-hosted] - Free, privacy-respecting metasearch. Point SEARXNG_URL at your instance.
(O) Tavily [paid] - Search + extract in one provider.
(O) xAI Web Search (Grok) [paid] - Agentic web search via Grok's web_search tool - uses xAI Grok OAuth or XAI_API_KEY.
(O) Skip - keep defaults / configure later

```

image4.png 49.4 KB

Will have to setup tools later as some are not connected

```

◆ Tool Availability Summary
5/9 tool categories available:

✓ Vision (image analysis)
✗ Web Search & Extract (missing EXA_API_KEY, PARALLEL_API_KEY, FIRECRAWL_API_KEY/FIRECRAWL_API_URL, TAVILY_API_KEY, or SEARXNG_URL)
✗ Browser Automation (missing npm install -g agent-browser && agent-browser install --with-deps)
✗ Image Generation (missing FAL_KEY or OPENAI_API_KEY)
✓ Text-to-Speech (Edge TTS)
✗ Skills Hub (GitHub) (missing GITHUB_TOKEN)
✓ Terminal/Commands
✓ Task Planning (todo)
✓ Skills (view, create, edit)

▲ Some tools are disabled. Run 'hermes setup tools' to configure them,
▲ or edit ~/.hermes/.env directly to add the missing API keys.

```

image14.png 38.1 KB

Enable browser

Install agent-browser

```
$ npm install -g agent-browser
```

Test

Start 'hermes' TUI

Create a test file

Prompt

Create a file /home/map7/foo.txt with bar in it

Then do a ls /home/map7 in terminal to check

Speed it up!

Using qwen/qwen3.5-9b from lmstudio-community model do the following through lmstudio

set context to 65000

turn off reasoning

turn off K & V Quantization

Install hermes-desktop and turn off reasoning in there, then try.

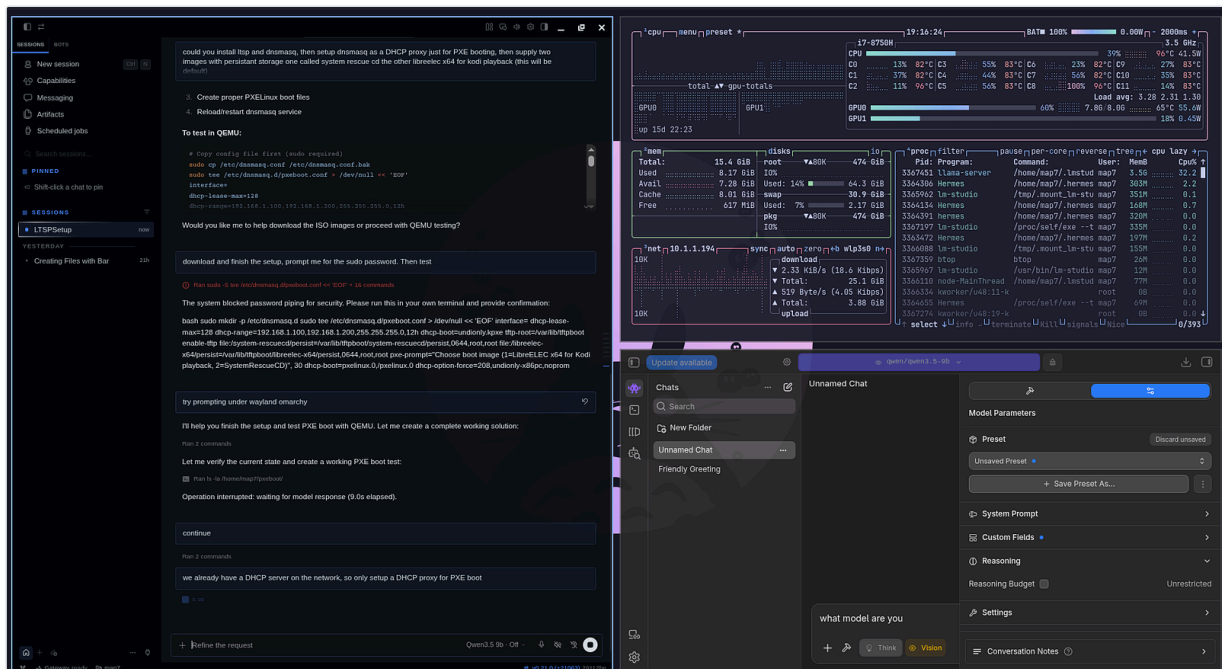


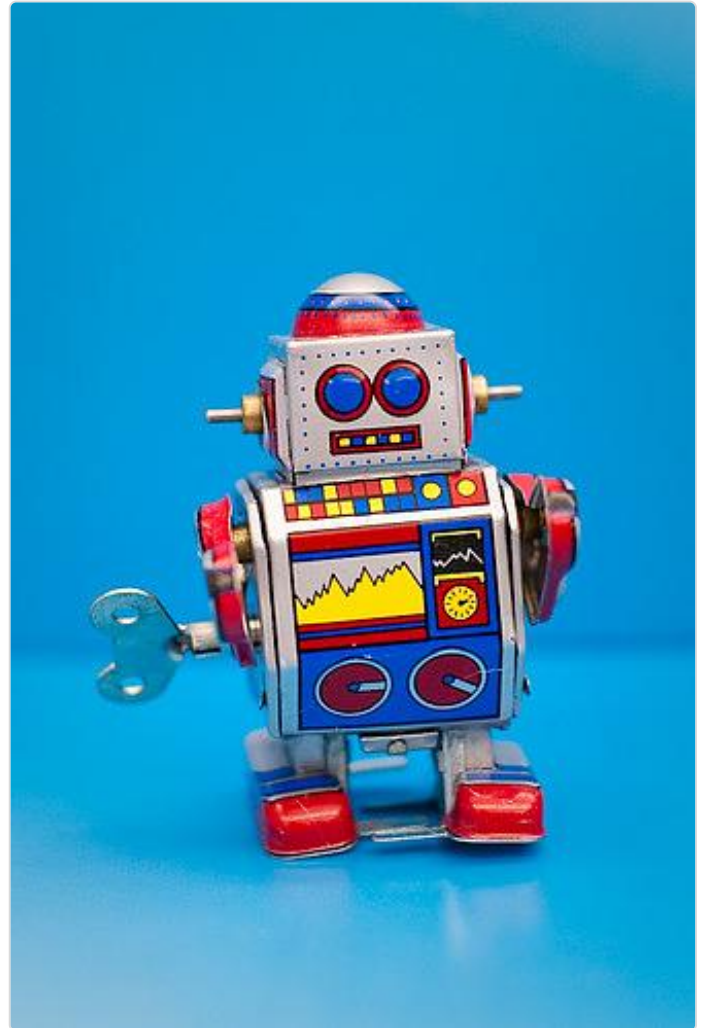
image.png 357 KB

t

Gallery



"Hermes Agent logo" by NousResearch
licensed under MIT.



"Windows 7 robot" by randychiu is licensed
under CC BY 2.0. To view a copy of this
license, visit
<https://creativecommons.org/licenses/by/2.0>