Installing Ollama on Windows 11

Local AI for the situation-report narration · default model mistral-nemo · ~15 minutes plus model download · no admin rights needed

1 Why local?

Ollama runs language models entirely on your own machine. For the situation report's AI narration this means: the delta briefing never leaves the system, and you need no account, no API key and no corporate approval for external services. This path shows up as deployment_class "local" in the AI banner and the operator-evidence log — as long as the address points to this machine (localhost, 127.0.0.1, ::1). If it points to another machine, the class reads "remote": the data leaves this machine, and whether your own network or the internet is behind it the tool cannot know. The log keeps the address as a SHA-256 hash.

2 Prerequisites

• Windows 11, 64-bit; no administrator rights needed (Ollama installs into the user profile).

• Disk space: ~4 GB for Ollama itself plus the model — mistral-nemo downloads about 7 GB, mistral about 4 GB.

• Memory: 16 GB recommended for mistral-nemo (12B); with 8 GB pick the smaller mistral (7B).

• Without a GPU everything runs on the CPU — a narration then takes up to a minute; an NVIDIA or AMD GPU speeds things up noticeably and is used automatically.

3 Install

• Open ollama.com/download/windows in the browser and download OllamaSetup.exe.

• Double-click the file and follow the wizard — there are no choices to make.

• Ollama then runs in the background; the llama icon appears in the taskbar's notification area. It starts automatically after a reboot.

Verify the installation — open PowerShell or the command prompt (Windows key, type "powershell"):

```
ollama --version
```

4 Pull the model

The narration's default model is mistral-nemo (12B, good multilingual quality, management tone). The download happens once:

```
ollama pull mistral-nemo

# Alternative fuer Rechner mit 8 GB RAM / for 8 GB machines:
ollama pull mistral
```

Quick test directly in Ollama — the first answer takes longest because the model is loaded into memory:

```
ollama run mistral-nemo "Antworte mit einem Satz: Was ist ein Lagebild?"
# Chat beenden / leave the chat: /bye
```

5 Wiring check with SituationReport

In the repository or release folder, the providers list shows the discovered AI backends, and the test command runs a first labeled sample narration through Ollama:

```
python -m llm providers
python -m llm models
python -m llm test
python -m llm test --lang en --model mistral
```

6 Choosing a model and an address

With more than one model in Ollama, the name decides which one runs. The models command lists what is actually installed and marks the default. Write the name exactly as it appears there — with its tag, so qwen3.8:latest rather than qwen3.8. If Ollama does not know the name, SituationReport says so and gives the

matching ollama pull command.

```
# Installierte Modelle auflisten / list installed models:
python -m llm models

# Ein bestimmtes Modell benutzen / use a specific model:
python -m llm test --model qwen3.8:latest

# Ollama auf einem anderen Rechner / Ollama on another host:
python -m llm models --base-url http://gpu-box:11434
```

In the GUI, a Model field and an Address field sit next to the provider. The model field offers the installed models but stays typable — a model still being pulled can be entered already. Empty means: let the provider decide. Both fields are remembered for the next start; they belong to THIS machine and therefore do NOT go into the saved solution configuration you hand to colleagues.

If Ollama runs elsewhere, name the address — in the Address field, via --base-url or --llm-base-url, or once via the OLLAMA_HOST environment variable that Ollama itself uses. Without one, http://localhost:11434 applies.

7 Use it

CLI: --narrate adds the "Narration (Entwurf)" section to the delta briefing — including the AI label, the numbers guard and the operator evidence (llm_audit.jsonl next to the output). GUI: in the Solutions & Portfolios window tick "AI narration (draft)", pick the provider next to it, then run "Delta briefing …" as usual:

```
python -m portfolio --delta prev.json now.json --narrate --output delta.html --browser

# Anderes Modell / different model:
python -m portfolio --delta prev.json now.json --narrate --llm-model mistral --output delta.html
```

8 Troubleshooting

• "Python was not found" (pointing to the Microsoft Store): Windows redirects the python command to a Store placeholder. Run the commands from sections 5 to 7 in the repository folder and use .venv\Scripts\python.exe there instead of python (or activate once via .venv\Scripts\Activate.ps1) — alternatively py -m llm test with the Python launcher installed, or disable the app execution aliases python.exe/python3.exe (Settings > Apps > Advanced app settings).

• "Could not reach Ollama" / connection refused: Ollama is not running — start "Ollama" from the Start menu and wait for the llama icon in the notification area.

• "Ollama does not know model …": the model is missing — run ollama pull mistral-nemo (section 4).

• Answers are slow: on a pure CPU up to a minute per narration is normal; mistral instead of mistral-nemo roughly halves the wait.

• Low space on C: — relocate the model folder:

```
# Modelle auf D: statt C: ablegen / store models on D:
setx OLLAMA_MODELS "D:\ollama-models"
# danach Ollama beenden und neu starten / then restart Ollama
```

• Quit: right-click the llama icon in the notification area → "Quit Ollama".

9 Privacy in one sentence

Ollama listens on your machine only (localhost:11434); neither briefing nor narration ever leaves the system — and the difference from the external path (Claude API) stays visible at all times via the deployment_class in banner and audit.