# llama.cpp 1.1.0: llama_batch deprecation, new quantization targets and server changes

27 November 2026 · 4 min read · blogs.iictp.org/llama-cpp-1-1-release-llama-batch-ext-upgrade

Short answer: llama.cpp 1.1.0 is a release of the open-source local inference engine, published on GitHub on 26 November 2026, that deprecates the old llama_batch API in favour of llama_batch_ext. It also adds new model families, size-targeted quantization and a reworked server scheduler. It matters because applications built on the C API should plan a migration and server users should check changed behaviour.

**TL;DR The short version**

- llama.cpp 1.1.0, released on 26 November 2026, deprecates the old llama_batch API in favour of llama_batch_ext.
- It also adds new model families, size-targeted quantization and a reworked server scheduler.
- Applications built on the C API should plan a migration, and server users should test changed behaviour before upgrading.

Read online, listen to it, or change the text size: https://blogs.iictp.org/llama-cpp-1-1-release-llama-batch-ext-upgrade

## What is llama.cpp 1.1.0?

llama.cpp runs language models in the GGUF file format on processors and GPUs, using the ggml tensor library. The release notes list added support for the Inkling 2 family, Domyn Edge 4B, Zaya1 8B, MiniMax-M3.1, RAON-1-mini, Hy4 flash and a set of speech and audio models, and an update of ggml to 1.1.1.

## API changes

- The llama_batch API is deprecated in favour of llama_batch_ext and llama_process.
- New functions identify the DFlash draft head variant and read string arrays from GGUF metadata.
- Quantization parameters gain a target bits-per-weight, a target size, a state file and an abort callback.
- The multimodal header adds audio types and related functions.

## What is new compared with 1.0

- Per-tensor quantization can be chosen automatically to meet --target-bpw or --target-size, with a resumable state file.
- DFlash speculative decoding is split into separate draft methods.
- llama-server has a refactored slot scheduler and clears slot cache when the global LoRA configuration changes.

- Chat parsing now matches special tokens as tokens instead of text.
- The web interface was redesigned, with voice messages and agent traces.

## Server behaviour reported in the release notes

- OpenAI-style named tool choice is accepted.
- A models event stream is added in single-model mode.
- Rejected reasoning effort levels, and grammar combined with ignore_eos, now return error 400.
- Empty stop strings are ignored.
- New flags include --ui-agent and --ui-mcp-proxy-allow-private-urls.
- The notes report a 63% cut in first-load JavaScript for the interface.

## Limitations and open questions

- The notes publish no general speed or accuracy benchmark for this version.
- Deprecated does not say when the old API will be removed.
- New model support in a first release can have defects; compare outputs with a reference implementation.
- The first version of the 1.x series is covered in our versioning guide.

Sources were read on 27 November 2026. Products, rules and data change, so check the linked originals before you rely on any detail.

## How to upgrade safely

1. Pin your current version and record outputs for a fixed prompt set.
2. Build 1.1.0 and treat deprecation warnings as a task list.
3. Port batching code to llama_batch_ext and test multi-sequence and multimodal paths.
4. Check clients for requests that now return 400.
5. Leave the private URL proxy flag off unless you need it, and restrict who can reach the server.
6. Compare size-targeted quantizations with your usual presets on your own tasks.

## What it means for local LLM engineers

Size-targeted quantization is useful when a model almost fits your memory, but it changes which tensors lose precision, so quality must be measured again. The server fixes around LoRA and cache handling deserve a regression test if you switch adapters at runtime.

## Frequently asked questions

- **What is deprecated in llama.cpp 1.1.0?** The old llama_batch API, in favour of llama_batch_ext.
- **Which new quantization options exist?** Targets for bits per weight or file size.
- **Does it add agent features?** The notes list agent traces in the interface and a flag for agent mode.
- **Should I upgrade now?** Test first, especially API code and server clients.

## Related reading

- llama.cpp 1.0 versioning
- LLM quantization: GGUF and AWQ
- Local LLM Practitioner certification

Related credential: Local LLM Practitioner. Explore the exam facts and syllabus.

### Sources and further reading

1. ggml-org. llama.cpp release v1.1.0. 26 November 2026. github.com/ggml-org/llama.cpp/releases/tag/v1.1.0

Published by IICTP, an independent technical credential body, on 27 November 2026. The current version of this post is at https://blogs.iictp.org/llama-cpp-1-1-release-llama-batch-ext-upgrade. More guides: blogs.iictp.org.