llama.cpp v1.0.0 explained: what the first stable release and API/ABI versioning mean

23 November 2026 · 6 min read · blogs.iictp.org/llama-cpp-v1-0-0-semantic-versioning-api-abi-explained

Short answer: on 13 November 2026 the ggml-org team announced v1.0.0 of the llama.cpp and ggml libraries, and said that from now on API and ABI incompatibilities will be signalled by major-version bumps. llama.cpp runs multimodal LLMs on hardware from consumer machines to enterprise servers through quantisation, optimised backends and a portable C/C++ implementation. For developers embedding it, version numbers now carry a compatibility promise.

TL;DR The short version

- llama.cpp has reached v1.0.0, and from now on breaking changes to its public C API and ABI will be signalled by major-version bumps.
- MINOR versions bring backward-compatible features, backends and model support, and the project supports more than 150 model architecture variants.
- Pin the major version in your build, and note that the policy covers the C API, not server flags or command-line tools.

Read online, listen to it, or change the text size: https://blogs.iictp.org/llama-cpp-v1-0-0-semantic-versioning-api-abi-explained

What is llama.cpp?

According to the post, written by Daniel Bevenius, Georgi Gerganov and merve of the llama.cpp team at Hugging Face, llama.cpp is an LLM inference library whose public header is libllama, built on ggml, a tensor library. It is 3.5 years old and many downstream applications depend on it.

How the new versioning works

- **MAJOR:** a breaking change to the public C API in the include directory.
- **MINOR:** a backward-compatible feature, backend update, model support or API addition.
- **PATCH:** a bug fix with no API change.
- The rules apply to all ggml-org projects using semantic versioning, and both llama.cpp and ggml are at v1.0.0.

What else the post says

- Distribution channels listed include pre-release artifacts, Docker, Debian, Homebrew, conda-forge, MacPorts, Winget and an XCFramework.

- More than 150 model architecture variants are supported, plus all mainstream speculative decoding methods, per the post.
- CI runs on self-hosted runners with hardware support from NVIDIA, AMD, Qualcomm, Arm, IBM and Intel, and uses HF Buckets and HF Jobs for capacity.
- On AI-assisted development, the team says LLMs are integrated “thoughtfully”, with code quality and architectural control as priorities.

Limitations and open questions

- The post is a project announcement; it does not publish performance data.
- It states a policy for the C API. Other interfaces, such as server flags and command-line tools, are not covered in what we read.
- The promise is as good as the project’s follow-through; check release notes.

How to adopt it, step by step

1. Pin the major version in your build files.
2. Read release notes for MINOR bumps, since backends and model support may change behaviour.
3. Test GGUF models and your sampling settings after each upgrade.
4. Subscribe to the project’s discussion on packaging if you redistribute it.

What it means for people running local models

A stable API reduces maintenance for apps that wrap llama.cpp. See LLM quantisation explained: GGUF and AWQ and hardware for local AI.

Frequently asked questions

- **What changed in v1.0.0?** The libraries adopted strict semantic versioning with major bumps for breaking API/ABI changes.
- **Does it support Apple, Windows and Linux?** The post lists distribution for Linux, macOS, Windows, Android and iOS.
- **How many architectures are supported?** More than 150 variants, per the post.

Related reading

- LLM quantisation explained: GGUF and AWQ
- Hardware for local AI
- Local LLM vs cloud AI
- Local LLM Practitioner

Sources and currency

Sources were read on 23 November 2026. Products, rules and standards change, so check the linked originals before you rely on any detail.

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

Sources and further reading

1. Daniel Bevenius, Georgi Gerganov, merve (ggml-org). llama.cpp v1.0.0: A stronger foundation for the local AI ecosystem. Hugging Face blog, 13 November 2026. huggingface.co/blog/ggml-org/llama-cpp-v1.

Published by IICTP, an independent technical credential body, on 23 November 2026. The current version of this post is at https://blogs.iictp.org/llama-cpp-v1-0-0-semantic-versioning-api-abi-explained. More guides: blogs.iictp.org.