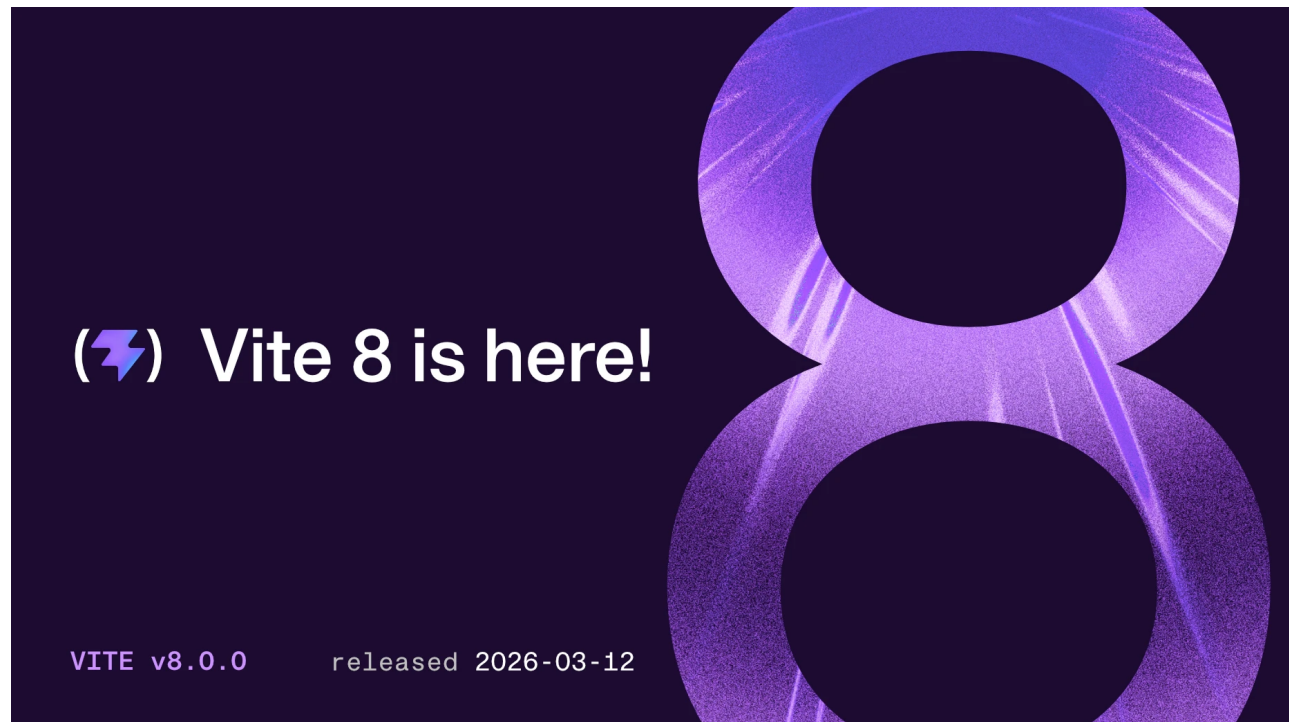




On this page

Vite 8.0 is out!

March 12, 2026



We're thrilled to announce the stable release of Vite 8! When Vite first launched, we made a pragmatic bet on two bundlers: esbuild for speed during development, and Rollup for optimized production builds. That bet served us well for years. We're very grateful to the Rollup and esbuild maintainers. Vite wouldn't have succeeded without them. Today, it resolves into one: Vite 8 ships with [Rolldown](#) as its single, unified, Rust-based bundler, delivering up to 10-30x faster builds while maintaining full plugin compatibility. This is the most significant architectural change since Vite 2.

Vite is now being downloaded 65 million times a week, and the ecosystem continues to grow with every release. To help developers navigate the ever-expanding plugin landscape, we also launched registry.vite.dev, a searchable directory of plugins for Vite, Rolldown, and Rollup that collects plugin data from npm daily.

Quick links:

- [Docs](#)
- Translations: [简体中文](#), [日本語](#), [Español](#), [Português](#), [한국어](#), [Deutsch](#), [فارسی](#)
- [Migration Guide](#)
- [GitHub Changelog](#)

Play online with Vite 8 using [vite.new](#) or scaffold a Vite app locally with your preferred framework running `pnpm create vite`. Check out the [Getting Started Guide](#) for more information.

We invite you to help us improve Vite (joining the more than [1.2K contributors to Vite Core](#)), our dependencies, or plugins and projects in the ecosystem. Learn more at our [Contributing Guide](#). A good way to get started is by [triaging issues](#), [reviewing PRs](#), sending tests PRs based on open issues, and supporting others in [Discussions](#) or Vite Land's [help forum](#). If you have questions, join our [Discord community](#) and talk to us in the [#contributing channel](#).

Stay updated and connect with others building on top of Vite by following us on [Bluesky](#), [X](#), or [Mastodon](#).

The Rollup-Powered Vite

The problem

Since its earliest versions, Vite relied on two separate bundlers to serve different needs. [esbuild](#) handled fast compilation during development (dependency pre-bundling and TypeScript/JCX transforms) that made the dev experience feel instant. [Rollup](#) handled production bundling, chunking, and optimization, with its rich plugin API powering the entire Vite plugin ecosystem.

This dual-bundler approach served Vite well for years. It allowed us to focus on developer experience and orchestration rather than reinventing parsing and bundling

from scratch. But it came with trade-offs. Two separate transformation pipelines meant two separate plugin systems, and an increasing amount of glue code needed to keep the two pipelines in sync. Edge cases around inconsistent module handling accumulated over time, and every alignment fix in one pipeline risked introducing differences in the other.

The solution

[Rolldown](#) is a Rust-based bundler built by the [VoidZero](#) team to address these challenges head-on. It was designed with three goals:

- **Performance:** Written in Rust, Rolldown operates at native speed. In benchmarks, it is [10-30x faster than Rollup](#) matching esbuild's performance level.
- **Compatibility:** Rolldown supports the same plugin API as Rollup and Vite. Most existing Vite plugins work out of the box with Vite 8.
- **Advanced features:** A single unified bundler unlocks capabilities that were difficult or impossible with the dual-bundler setup, including full bundle mode, more flexible chunk splitting, module-level persistent caching, and Module Federation support.

The journey to stable

The migration to Rolldown was deliberate and community-driven. First, a separate [rolldown-vite](#) package was released as a technical preview, allowing early adopters to test Rolldown's integration without affecting the stable version of Vite. The feedback from those early adopters was invaluable. They pushed the integration through real-world codebases of every shape and size, surfacing edge cases and compatibility issues we could address before a wider release. We also set up a dedicated CI suite validating key Vite plugins and frameworks against the new bundler, catching regressions early and building confidence in the migration path.

In December 2025, we shipped the [Vite 8 beta](#) with Rolldown fully integrated. During the beta period, Rolldown itself progressed from beta to a release candidate, with continuous improvements driven by the testing and feedback of the Vite community.

Real-world performance

During the preview and beta phases of `rolldown-vite`, several companies reported measurable reductions in production build times:

- **Linear:** Production build times dropped from 46s to 6s
- **Ramp:** 57% build time reduction
- **Mercedes-Benz.io:** Up to 38% build time reduction
- **Beehiiv:** 64% build time reduction

For large projects, the impact can be especially noticeable, and we expect further improvements as Rolldown continues to evolve.

A unified toolchain

With Vite 8, Vite becomes the entry point to an end-to-end toolchain with closely collaborating teams: the build tool (Vite), the bundler (Rolldown), and the compiler ([Oxc](#)). This alignment ensures consistent behavior across the entire stack, from parsing and resolving to transforming and minifying. It also means we can rapidly adopt new language specifications as JavaScript evolves. And by integrating deeply across layers, we can pursue optimizations that were previously out of reach, such as leveraging Oxc's semantic analysis for better tree-shaking in Rolldown.

Thank you to the community

None of this would have been possible without the broader community. We want to extend our deep thanks to the framework teams ([SvelteKit](#), [React Router](#), [Storybook](#), [Astro](#), [Nuxt](#), and many others) who tested `rolldown-vite` early, filed detailed bug reports, and worked with us to resolve compatibility issues. We are equally grateful to every developer who tried the beta, shared their build time improvements, and reported the rough edges that helped us polish this release. Your willingness to test the migration on real projects helped make the transition to Rolldown smoother and more reliable.

Node.js Support

Vite 8 requires Node.js 20.19+, 22.12+, the same requirements as Vite 7. These ranges ensure Node.js supports `require(esm)` without a flag, allowing Vite to be distributed as ESM only.

Additional Features

Beyond the Rolldown integration, Vite 8 includes several notable features:

- **Integrated Devtools:** Vite 8 ships `devtools` option to enable [Vite Devtools](#), a developer tooling for debugging and analysis. Vite Devtools provide deeper insights into your Vite-powered projects directly from the dev server.
 - **Built-in `tsconfig paths` support:** Developers can enable TypeScript path alias resolution by setting `resolve.tsconfigPaths` to `true`. This has a small performance cost and is not enabled by default.
 - **`emitDecoratorMetadata` support:** Vite 8 now has built-in automatic support for TypeScript's `emitDecoratorMetadata` option, removing the need for external plugins. See the [Features](#) page for details.
 - **Wasm SSR support:** `.wasm?init imports` now work in SSR environments, expanding Vite's WebAssembly feature to server-side rendering.
 - **Browser console forwarding:** Vite 8 can forward browser console logs and errors to the dev server terminal. This is especially useful when working with coding agents, as runtime client errors become visible in the CLI output. Enable it with `server.forwardConsole`, which activates automatically when a coding agent is detected.
-

@vitejs/plugin-react v6

Alongside Vite 8, we are releasing `@vitejs/plugin-react` v6. The plugin uses Oxc for React Refresh transform. Babel is no longer a dependency and the installation size is smaller.

For projects that need the [React Compiler](#), v6 provides a `reactCompilerPreset` helper that works with `@rollup/plugin-babel`, giving you an explicit opt-in path without burdening the default setup.

See [the Release Notes](#) for more details.

Note that v5 still works with Vite 8, so you can upgrade the plugin after upgrading Vite.

Looking Ahead

The Rolldown integration opens the door to improvements and optimizations. Here is what we are working on next:

- **Full Bundle Mode** (experimental): This mode bundles modules during development, similar to production builds. Preliminary results show 3x faster dev server startup, 40% faster full reloads, and 10x fewer network requests. This is especially impactful for large projects where the unbundled dev approach hits scaling limits.
- **Raw AST transfer**: Allows JavaScript plugins to access the Rust-produced AST with minimal serialization overhead, bridging the performance gap between Rust internals and JS plugin code.
- **Native MagicString transforms**: Enables custom transforms where the logic lives in JavaScript but the string manipulation computation runs in Rust.
- **Stabilizing the Environment API**: We are working to make the Environment API stable. The ecosystem has started regular meetings to better collaborate together.

Install Size

We want to be transparent about changes to Vite's install size. Vite 8 is approximately 15 MB larger than Vite 7 on its own. This comes from two main sources:

- **~10 MB from lightningcss:** Previously an optional peer dependency, lightningcss is now a normal dependency to provide better CSS minification out of the box.
- **~5 MB from Rolldown:** The Rolldown binary is larger than esbuild + Rollup mainly due to performance optimizations that favor speed over binary size.

We will continue monitoring and working to reduce install size as Rolldown matures.

Migrating to Vite 8

For most projects, upgrading to Vite 8 should be a smooth process. We built a compatibility layer that auto-converts existing `esbuild` and `rollupOptions` configuration to their Rolldown and Oxc equivalents, so many projects will work without any config changes.

For larger or more complex projects, we recommend the gradual migration path: first switch from `vite` to the `rolldown-vite` package on Vite 7 to isolate any Rolldown-specific issues, then upgrade to Vite 8. This two-step approach makes it easy to identify whether any issues come from the bundler change or from other Vite 8 changes.

Please review the detailed [Migration Guide](#) before upgrading. The complete list of changes is in the [Vite 8 Changelog](#).

Thank You, Rollup and esbuild

As Vite moves to Rolldown, we want to take a moment to express our deep gratitude to the two projects that made Vite possible.

Rollup has been Vite's production bundler since the very beginning. Its elegant plugin API design proved so well-conceived that Rolldown adopted it as its own, and Vite's entire plugin ecosystem exists because of the foundation Rollup laid. The quality and thoughtfulness of Rollup's architecture shaped how Vite thinks about extensibility. Thank you, [Rich Harris](#) for creating Rollup, and [Lukas Taegert-Atkinson](#) and the Rollup team for maintaining and evolving it into something that has had such a lasting impact on the web tooling ecosystem.

esbuild powered Vite's remarkably fast development experience from its early days: dependency pre-bundling, TypeScript and JSX transforms that completed in milliseconds rather than hundreds. esbuild proved that build tools could be orders of magnitude faster, and its speed set the bar that inspired an entire generation of Rust and Go-based tooling. Thank you, [Evan Wallace](#), for showing all of us what was possible.

Without these two projects, Vite would not exist as it does today. Even as we move forward with Rolldown, the influence of Rollup and esbuild is deeply embedded in Vite's DNA, and we are grateful for everything they have given to the ecosystem. You can learn more about all the projects and people Vite depends on at our [Acknowledgements](#) page.

Acknowledgments

Vite 8 was led by [sapphi-red](#) and the [Vite Team](#) with the help of the wide community of contributors, downstream maintainers, and plugin authors. We want to thank the [Rolldown team](#) for their close collaboration in making the Rolldown-powered Vite 8 possible. We are also especially grateful to everyone who participated in the `rolldown-vite` preview and the Vite 8 beta period. Your testing, bug reports, and feedback made the Rolldown migration possible and shaped this release into something we are proud of.

Vite is brought to you by [VoidZero](#), in partnership with [Bolt](#) and [NuxtLabs](#). We also want to thank our sponsors on [Vite's GitHub Sponsors](#) and [Vite's Open Collective](#).

[Suggest changes to this page](#)

© 2019-present VoidZero Inc. and Vite contributors. (faeb7467)