HEADS UP This site is currently under heavy development.

THE AI TOOLCHAIN NO. 001

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED AUGUST 19, 2026 · EVERY WEEKDAY

EDITIONS **tail** | grep | head | diff | uniq

Every release worth knowing about, the day it ships.

// HOW THIS ISSUE IS MADE

We read every release from the 119 tools on our watchlist at the source — **GitHub and GitLab release notes**, vendor **release pages** and **changelogs**, project **blogs and feeds**, vendor **press releases**, and the **source code** behind the tag. Bug-fix-only releases and non-product newsroom noise are dropped; what's left is summarized down to the new capability, how to try it, and any **screenshots or videos** the release itself published. Every entry links to the sources it was built from.

// CONTENTS

AI & LLM TOOLING

Local LLM Runtimes	vMLX

AI & LLM TOOLING

◆ AI Coding Agents

OpenAI Codex CLI ⓘ 43 RELEASES · 2026-07-21 → 2026-08-19 NOTES ↗

CODE ↗

Lightweight coding agent that runs in your terminal

> Across 43 releases, Codex CLI added Amazon Bedrock as a built-in provider, a Guardian V2 risk-scoring approval pipeline, an Agent Plugins marketplace with portable manifests, full MCP 2026 protocol support, and a broad TUI overhaul covering session forking, Markdown export, thread pinning and paginated history — alongside extensive sandbox, network-approval, and `codex doctor` diagnostics hardening.

└──▷ WHAT SHIPPED · 49 FEATURES most completely described first ? what's the number?

01 Session forking, archiving, pinning and naming NEW 87

Adds `codex exec fork` to fork a previous session by UUID or thread name into a new session, with an optional `--image` / `-i` attachment flag and optional prompt argument; adds archive and restore actions to the TUI resume picker; names new sessions with `/new` or `/clear`, and allows naming forked chats; adds `isPinned` on thread responses with `thread/metadata/update` to pin/unpin threads and an `isPinned` filter on `thread/list` with cursor-based pagination; supports `excludeTurns: true` in `thread/fork` to create ephemeral forks not exposed via `thread/list`.

Fork an existing session to explore an alternative approach without losing the original conversation state.

```
$ codex exec fork
```

Fork an existing session by ID and immediately send a follow-up prompt to continue work on a new branch of the conversation.

```
$ codex exec fork session-123 'continue refactoring the auth module'
```

— Full command, flags and API fields given `rust-v0.148.0` `rust-v0.148.0-alpha.5` `rust-v0.148.0-alpha.2` `rust-v0.147.0-alpha.13` `rust-v0.146.0` `rust-v0.146.0-alpha.8`

02 Guardian V2 risk-based approval review NEW 83

`features.guardianv2` now accepts either a boolean toggle or a structured object specifying classifier instructions, review threshold, reasoning effort, action/instruction token limits, transcript source controls, per-entry and total token budgets, and the number of recent non-user entries; adds per-thread risk samplers recording risk-score snapshots and requiring automatic review for high-risk actions; isolates Guardian reviewer sessions from parent extensions; enforces strict auto-review for MCP tool calls even when approval policy, tool annotations, or a remembered decision would otherwise skip review; gives Guardian full tool action context (transcript plus a structured planned-action object with tool name and arguments) and includes `node_repl` images and policy approval reasons in review evidence; adds a circuit-breaker that interrupts turns after the first Guardian denial for models whose catalog specialty is `cyber`; switches API-key session reviews to the `gpt-5.6-luna` model (retaining `codex-auto-review` for ChatGPT auth) routed through Responses Lite with fallback to the bundled Guardian policy.

— Names config key and models; lacks exact CLI invocation `rust-v0.149.0-alpha.1`

`rust-v0.148.0-alpha.23` `rust-v0.148.0-alpha.18` `rust-v0.148.0-alpha.16`
`rust-v0.148.0-alpha.14` `rust-v0.148.0-alpha.6` `rust-v0.148.0-alpha.1`
`rust-v0.147.0-alpha.11` `rust-v0.146.0-alpha.10`

---

03 ### `codex doctor` diagnostics expansion IMPROVED 82

Expands `codex doctor` with desktop app diagnostics, desktop update diagnostics, and desktop security enforcement diagnostics (app-server handshake status, macOS Gatekeeper/XProtect and Windows enforcement, update state); adds dedicated network diagnostics via a new `network` module covering proxy environment details, macOS system proxy detection, and rejection of proxy auth challenges (HTTP 407); adds sandbox checks via new `sandbox_check`; adds storage diagnostics reporting free space for `CODEX_HOME` and the worktree with a 5 GiB warning and 1 GiB failure threshold, plus Windows Dev Drive health checks.

— Names command and thresholds; explains mechanism per check `rust-v0.149.0-alpha.1`

`rust-v0.148.0-alpha.22` `rust-v0.148.0-alpha.21` `rust-v0.148.0-alpha.20`

---

04 ### `migrate-rollouts` subcommand NEW 80

Adds `migrate-rollouts` with `--apply`, `--thread <THREAD_ID>`, `--max-mib-per-second <MIB>`, `--json`, and `--verbose` flags to inspect or migrate legacy local sessions to paginated thread history.

Inspect or migrate legacy local sessions to paginated thread history before resuming old conversations.

```
$ codex migrate-rollouts
```

Dry-run to see which local sessions are eligible for migration to paginated thread history before committing.

```
$ codex migrate-rollouts --verbose --json
```

Migrate a specific thread to paginated history with a bandwidth cap to avoid saturating disk I/O.

```
$ codex migrate-rollouts --apply --thread <THREAD_ID> --max-mib-per-second 50
```

— Complete flag list with runnable examples `rust-v0.148.0-alpha.5` `rust-v0.148.0-alpha.1`

05 **Code Mode remote host support** NEW 79

Adds `--code-mode-host` accepting a `wss://` WebSocket URL (invalid schemes like `http://` or bare `ws://` are rejected at parse time) to connect the app server to a remote code-mode host over WebSocket; adds a dual-WebSocket transport routing frames via transport_lane(); supports gRPC code-mode hosts with automatic reconnection after restarts and removes the gRPC code-mode open session limit so more than `MAX_IN_FLIGHT_REQUESTS` concurrent sessions are allowed; disables Nagle's algorithm for code-mode WebSockets; adds a `features.code_mode_host` config table with a `disable_in_process_fallback` key (still accepts a plain boolean), a `code_mode_tool_names` field in Responses Lite turn metadata, and timeout handling for stalled host requests.

Connect the app server to a remote code-mode host over a secure WebSocket instead of the default local transport.

```
$ codex-app-server --code-mode-host wss://example.com/code-mode --listen stdio://
```

— Exact flag and URL scheme with example `rust-v0.148.0-alpha.18` `rust-v0.148.0-alpha.11` `rust-v0.148.0-alpha.6` `rust-v0.148.0-alpha.4` `rust-v0.147.0-alpha.7` `rust-v0.146.0` `rust-v0.146.0-alpha.10` `rust-v0.146.0-alpha.6`

06 **exec-server subcommand and remote execution** NEW 79

Adds `ExecServerSubcommand` accessible as `codex exec-server <subcommand>`, a `--remote <URL>` flag (marked `global`) for registering the exec-server as a remote environment, and a global `--strict-config`; propagates W3C `traceparent` / `tracestate` trace context through exec-server relay frames, including fragmented encrypted Noise-protocol requests; adds `--concurrent-requests`

(default `1`) for opt-in concurrent request dispatch; reports temporary directories in exec-server environment info, routes exec-server HTTP/WebSocket traffic through the configured proxy policy, and adds executor-local config layer loading.

Run the exec-server with up to 4 concurrent requests per connection to increase throughput in high-load environments.

```
$ codex exec-server --concurrent-requests 4 --listen
ws://127.0.0.1:8080
```

— Flags and subcommand named with example `rust-v0.148.0-alpha.23` `rust-v0.148.0-alpha.20` `rust-v0.148.0-alpha.6` `rust-v0.148.0-alpha.4` `rust-v0.148.0-alpha.2` `rust-v0.147.0-alpha.10`

---

07 **Amazon Bedrock Runtime provider support** NEW 78

Adds the built-in `amazon-bedrock-runtime` provider for regional OpenAI-compatible Bedrock endpoints with SigV4 auth, per-provider AWS profile/region/transport overrides, global and US cross-region model variants, and GPT-5.6 routing; adds experimental Amazon Bedrock login with managed authentication, custom endpoint configuration, and GPT-5.6 Sol as the default Bedrock model; enables cached web search (advertising hosted text web search while marking external live/indexed access unsupported, falling back to cached search or disabling the tool when managed requirements prohibit it) and remote conversation compaction; normalizes Bedrock model catalogs to text-only web search payloads by removing the `search_content_types` field; supports custom transports for non-default network/proxy configurations.

— Names provider, auth method, fallback logic; no exact enable command

`rust-v0.148.0-alpha.15` `rust-v0.147.0-alpha.6.6` `rust-v0.148.0-alpha.6` `rust-v0.147.0` `rust-v0.147.0-alpha.6.5` `rust-v0.147.0-alpha.10` `rust-v0.147.0-alpha.6.4` `rust-v0.145.0` `rust-v0.145.0-alpha.28`

---

08 **MCP OAuth and network hardening** IMPROVED 75

Adds `oauth.callback_port` to MCP server configuration so each server can override the global `mcp_oauth_callback_port` for CLI login, app-server, plugin installation, executor, and skill dependency OAuth flows; routes MCP OAuth discovery and auth status through each server's runtime HTTP client, capping local discovery at 5 seconds while preserving explicit login timeouts; isolates MCP OAuth credentials by environment and restricts hosted MCP credentials to local environments; distinguishes unknown MCP authentication status as a separate state; scopes MCP resource reads by connector to prevent cross-connector data access; restricts MCP HTTP redirects to the configured origin and isolates MCP resource headers during OAuth requests.

Pin a specific MCP server to its own OAuth callback port so it does not conflict with other servers sharing the global port.

```yaml
oauth:
  callback_port: 9876
```

— Names config key and timeout; behavior well explained `rust-v0.149.0-alpha.1`

`rust-v0.148.0-alpha.23` `rust-v0.148.0-alpha.22` `rust-v0.148.0-alpha.14`
`rust-v0.147.0-alpha.13` `rust-v0.147.0-alpha.2` `rust-v0.147.0-alpha.1`
`rust-v0.146.0-alpha.8`

---

09 **Agent Plugins marketplace and portable plugins** NEW 74

Introduces installable portable Agent Plugins with search across local, personal, workspace, and remote plugin catalogs; supports the Agent Plugins 1.0 schema, recognizing root `plugin.json` files and mapping metadata, `skills/`, and `mcp.json` into Codex plugin manifests, with `.codex-plugin/plugin.json` as a fallback overlay; adds workspace plugin publishing and marketplaces for Amazon Bedrock and Claude Code, `canPublishToWorkspace` metadata on plugin share contexts and `plugin/share/save` responses, and publishing through share updates; caches remote plugin catalogs by scope, infers the bundled Claude Code marketplace automatically, and uses the API marketplace for Amazon Bedrock; fetches remote installed plugins across all scopes; skips symbolic links when copying/installing a plugin into the store so symlinked skill files or executables no longer fail installation.

— Names manifest files and fields; enablement path unclear `rust-v0.147.0-alpha.6.6`

`rust-v0.148.0-alpha.5` `rust-v0.148.0-alpha.1` `rust-v0.147.0` `rust-v0.147.0-alpha.6.5`
`rust-v0.147.0-alpha.10` `rust-v0.146.0` `rust-v0.146.0-alpha.8` `rust-v0.146.0-alpha.7`
`rust-v0.146.0-alpha.5` `rust-v0.145.0-alpha.28`

10 **Skill catalog and executor skill access** IMPROVED 74

Adds a configurable skill catalog token budget; simplifies `skills.read` so `package` is the only required argument, resolving the owning catalog automatically and defaulting omitted `resource` to the package's main `SKILL.md`; lets executor skill packages be passed directly to `skills.read` without a preceding `skills.list` lookup, including skill root aliases; honors per-directory bundled skill settings in `skills/list` requests; moves the host skills service into the skills extension with host skill root loading; generalizes skill locator aliases across providers and aliases resource-backed skill locators under context pressure; exposes executor skills through skill tools with resource reads for explicitly selected skills.

– Names API methods and files; usage flow clear `rust-v0.149.0-alpha.1`

`rust-v0.148.0-alpha.21` `rust-v0.148.0-alpha.11` `rust-v0.148.0-alpha.7` `rust-v0.148.0-alpha.6` `rust-v0.148.0-alpha.4` `rust-v0.147.0-alpha.1` `rust-v0.146.0-alpha.7`

11 **Shell command and environment policy hardening** IMPROVED 74

Adds a fail-closed Tree-sitter-based PowerShell lowerer that converts a conservative subset of literal PowerShell commands into argument vectors, rejecting dynamic expressions, `#requires` directives, and unrecognized syntax; enforces environment-specific command policies and per-environment shell variable policies; adds a per-environment `allow_login_shell` policy exposing the `login` argument for shell tools when any selected environment permits login shells; requires approval for commands with dynamic shell words and requires fresh approval beneath denied permission paths; restricts `shell_command` to a single local environment; enforces non-interactive approval policy for Codex delegate sessions.

– Names config keys and args; policy scope explained `rust-v0.149.0-alpha.1`

`rust-v0.148.0-alpha.23` `rust-v0.148.0-alpha.22` `rust-v0.148.0-alpha.11` `rust-v0.147.0-alpha.13` `rust-v0.147.0-alpha.7`

12 **TypeScript SDK `configOverrides`** NEW 74

Adds `CodexOptions.configOverrides` to the TypeScript SDK for passing ordered `--config key=value` arguments to Codex CLI unchanged, enabling TOML permission maps with literal path keys that the structured config API cannot represent.

Pass a raw TOML permission-map override with a literal path key via the TypeScript SDK when the structured config API cannot represent it.

```javascript
const codex = new Codex({
  configOverrides: [
    '--config', 'sandbox.writable_roots."/var/myapp/data"=true'
  ]
});
```

— Named SDK field with code example `rust-v0.148.0-alpha.21`

---

13 **Authentication and agent identity hardening** IMPROVED

74

Adds provider-owned authentication recovery so providers can handle their own auth refresh flow independently; integrates workload identity with Codex authentication; adds local `requirements.toml` allowlists for login methods and ChatGPT workspaces enforced before stored or environment-provided credentials are used, including during bootstrap before cloud requirements are fetched; supports `CODEX_AGENT_IDENTITY_AUTHAPI_BASE_URL` and `CODEX_AGENT_IDENTITY_JWKS_BASE_URL` to override agent identity registration and verification endpoints.

Point agent identity registration and verification at a staging environment instead of production.

```
$ CODEX_AGENT_IDENTITY_AUTHAPI_BASE_URL=https://staging-
auth.example.com
CODEX_AGENT_IDENTITY_JWKS_BASE_URL=https://staging-
jwks.example.com codex
```

— Named env vars with example and scope `rust-v0.148.0-alpha.23` `rust-v0.148.0-alpha.11` `rust-v0.148.0-alpha.2` `rust-v0.147.0-alpha.12`

---

14 **Thread cost and credit visibility** NEW

73

Adds `thread-credits` and `estimated-thread-cost` as configurable terminal title items and shows estimated thread usage/cost in the `/status` command and TUI status surfaces for eligible workspaces, alongside new per-thread usage queries to the backend client.

Check estimated credit usage or cost for the current thread directly in the TUI.

```
$ /status
```

Surface per-thread credit consumption and estimated cost directly in your terminal title bar while running Codex.

```yaml
thread-credits, estimated-thread-cost
```

— Names config keys and command; easy to try `rust-v0.148.0` `rust-v0.148.0-alpha.11`

---

15 **Installer source selection via `CODEX_INSTALLER_USE_RELEASES_OPENAI_COM`**

NEW 72

Adds `CODEX_INSTALLER_USE_RELEASES_OPENAI_COM` to control whether the installer pulls from `https://releases.openai.com/codex` or falls back to GitHub Releases; standalone installers now prefer `releases.openai.com` by default, with the variable set to `false` forcing GitHub Releases.

Force the Codex installer to use GitHub Releases instead of `releases.openai.com`, useful when the OpenAI CDN is unreachable in a restricted environment.

```
$ curl -fsSL https://chatgpt.com/codex/install.sh |
CODEX_INSTALLER_USE_RELEASES_OPENAI_COM=false sh
```

Force the standalone installer to use GitHub Releases instead of `releases.openai.com`, useful in environments that block the primary CDN.

```
$ curl -fsSL https://chatgpt.com/codex/install.sh |
CODEX_INSTALLER_USE_RELEASES_OPENAI_COM=false sh
```

Force the Codex installer to use GitHub Releases instead of `releases.openai.com` when behind a network that blocks the default source.

```
$ curl -fsSL https://chatgpt.com/codex/install.sh |
CODEX_INSTALLER_USE_RELEASES_OPENAI_COM=false sh
```

— Named env var with runnable examples `rust-v0.146.0` `rust-v0.146.0-alpha.5` `rust-v0.145.0-alpha.29`

---

16 **Sandbox filesystem and path-safety hardening** IMPROVED

71

Adds symlink-safe reading of sensitive files via `read_sensitive_file_to_string`, refusing to follow the final symlink component on Unix or reparse points on Windows; rejects symlinks used as memory workspace roots and removes links created during consolidation; isolates external editor buffers from sandbox-writable paths; drops capabilities from Linux sandbox processes and hardens Windows sandbox provisioning against reparse points; mounts a minimal `/dev` in full-filesystem Bubblewrap sandboxes; makes Windows `PathUri` equality/hashing ASCII-case-insensitive for drive and UNC paths while preserving case-sensitive POSIX behavior; validates images before returning `view_image` output; redacts secrets from app-server command execution items and improves bearer token redaction.

— Names internal functions; mostly non-actionable hardening `rust-v0.149.0-alpha.1`

`rust-v0.148.0-alpha.23` `rust-v0.148.0-alpha.22` `rust-v0.148.0-alpha.7` `rust-v0.148.0-alpha.6` `rust-v0.148.0-alpha.5` `rust-v0.148.0-alpha.1` `rust-v0.147.0-alpha.12`

---

17 ### `/export` Markdown conversation export NEW

71

Adds an `/export` TUI command that exports the full conversation history to Markdown, either to the clipboard or a new file, with a default filename prompt and support for an explicit destination path argument.

Export your current TUI conversation to a Markdown file to share or archive a debugging session.

```
$ /export
```

Export the current TUI conversation to a Markdown file for documentation or sharing.

```
$ /export ~/projects/my-session.md
```

— Runnable command with destinations documented `rust-v0.148.0` `rust-v0.148.0-alpha.2`

---

18 ### `/import` migration from Cursor and Claude Code IMPROVED

70

Expands `/import` to migrate Cursor and Claude Code settings, MCP servers, plugins, sessions, commands, and project-scoped memories, and to import Cursor-managed skills while syncing imported Claude/Cursor conversations without duplicates; bounds Cursor project path resolution to 128 path candidates, rejecting ambiguous matches and unsafe encoded components; preserves timestamps when importing external agent sessions and detects connectors used in external agent sessions; adds `maxSessionAgeDays` and `maxSessions` fields to the external-agent config detection

request (defaults 30 days, 50 sessions) plus an optional `providerId` field on `externalAgentConfig/import` for analytics attribution.

– Names command and fields; clear entry point `rust-v0.147.0-alpha.6.6` `rust-v0.147.0` `rust-v0.147.0-alpha.13` `rust-v0.146.0-alpha.5` `rust-v0.145.0` `rust-v0.145.0-alpha.29`

---

19 **`--approve-for-me` auto-reviewed approvals** NEW 68

Adds `--approve-for-me` to `codex` to enable automatically reviewed approvals without manual confirmation prompts.

Run a coding task in CI without manual approval prompts by enabling auto-reviewed approvals.

```
$ codex --approve-for-me 'refactor the auth module to use async/await'
```

– Flag with runnable example and clear effect `rust-v0.147.0`

---

20 **Unified network approval pipeline** NEW 67

Routes blocked network requests through the shared approval pipeline so permission hooks, automatic review, and user review all use the common flow, including for background terminals started by an earlier turn; persists deny amendments so denied requests stay denied across turns, and records the final applied network decision in telemetry without exposing the destination; scopes pending approvals to a turn and execution, coalesces duplicate requests within one execution, and cancels in-flight Guardian reviews when their owner is dropped; denies network access outright when an allow amendment fails; adds network policy metadata to environment configuration.

– Explains mechanism thoroughly; no direct user command `rust-v0.148.0-alpha.22` `rust-v0.148.0-alpha.12` `rust-v0.147.0-alpha.13` `rust-v0.146.0-alpha.10`

21 **Paginated thread history** NEW 65

Adds experimental paginated thread history with efficient resume, search, persisted names, sub-agent support, and memories; adds `includeTurns` reads for incremental transcript access and durable reverts for paginated threads; uses paginated history for persistent exec threads to keep memory bounded across long sessions; supports paginated thread forks and paginated transcript loading in the TUI for large histories; speeds up `exec resume --last` by querying the state database first, falling back to a full rollout scan only on a complete miss, and prefers SQLite-backed names when resolving local `archive`, `delete`, and `unarchive` targets.

— Explains scope and behavior; not a single command `rust-v0.148.0-alpha.20`

`rust-v0.148.0-alpha.11` `rust-v0.148.0-alpha.6` `rust-v0.147.0-alpha.10` `rust-v0.147.0-alpha.7` `rust-v0.146.0-alpha.7` `rust-v0.145.0`

22 **MCP tool discovery and namespace management** IMPROVED 65

Instructs the model to use `tool_search` to discover relevant tools when an explicitly selected plugin has apps available, before falling back to built-in tools; reuses pending MCP server connections during reconciliation when identity, catalog limit, and protocol mode match; caches tool catalogs for streamable HTTP MCP servers and speeds up MCP OAuth credential reads; exposes plugin ownership in MCP server status; supports deferred loading for freeform and custom tools in tool search and custom tools in namespaces; adds tool registry collision policy configuration (`#36954`) enforcing strict tool name collision errors under a canonical `functions` namespace; includes tool namespace inventory in turn metadata and adds dynamic HTTP header helpers for MCP servers.

— Names PR and namespace; mostly internal behavior `rust-v0.148.0-alpha.16`

`rust-v0.148.0-alpha.11` `rust-v0.148.0-alpha.7` `rust-v0.148.0-alpha.6` `rust-v0.148.0-alpha.4` `rust-v0.147.0-alpha.10`

23 **`codex agents` dashboard** NEW 65

Adds `codex agents` as a dedicated interactive task dashboard command for monitoring and navigating agents in the TUI, with configurable keyboard shortcuts.

Open the interactive agents dashboard to monitor and navigate running agent tasks in the TUI.

```
$ codex agents
```

— Runnable command with dashboard entry point `rust-v0.148.0-alpha.22`

24 **`--full-auto` flag removed** BREAKING 65

Removes the `codex exec --full-auto` flag; use `--sandbox workspace-write` as the supported equivalent.

Run a sandboxed exec task using the replacement for the removed `--full-auto` flag.

```
$ codex exec --sandbox workspace-write 'add unit tests for the payments service'
```

— Names both flags with migration example `rust-v0.147.0` `rust-v0.147.0-alpha.13`

---

25 **Interrupted turn recovery via `RecoverTurnRequest`** NEW 64

Adds `RecoverTurnRequest` and `CodexThread::recover_turn_if_idle` to resume an interrupted regular turn using its existing turn ID and updated thread settings, including Plan-mode recovery without injecting an empty user message.

— Names internal API; not a user-facing action `rust-v0.148.0-alpha.12`

---

26 **Multi-agent v2 support** IMPROVED 62

Adds a nullable `multiAgentVersion` field to `v2` `model/list` responses with supported values `disabled`, `v1`, or `v2`; adds leaf model support in multi-agent v2 and tracks context windows per agent; adds configurable developer instructions for v2 subagents; allows disabling the multi-agent wait tool and suppresses empty multi-agent mode messages in the TUI; identifies agents by name in token budget context with configurable identity; deduplicates current-time reminders when spawning full-history subagents, replacing inherited parent reminders with a single fresh reminder.

— Names field and values; no direct user action `rust-v0.148.0-alpha.14`

`rust-v0.148.0-alpha.6` `rust-v0.148.0-alpha.5` `rust-v0.148.0-alpha.4`
`rust-v0.147.0-alpha.7` `rust-v0.147.0-alpha.2` `rust-v0.147.0-alpha.1`
`rust-v0.146.0-alpha.5` `rust-v0.145.0-alpha.28`

THINNER COVERAGE BELOW

27 **MCP 2026 protocol support and conformance gates** NEW 58

Adds support for the opt-in MCP 2026-07-28 protocol, including paginated discovery, multi-round requests, and non-blocking server startup, completing full MCP 2026 client support; raises the MCP server recursion limit and exposes cached MCP tools before the server finishes starting up; adds MCP client conformance regression gates covering HTTP and stdio transports, OAuth scenarios, protocol versions, and an app-server

regression matrix for transport, security, schema, pagination, SSE, multi-round, and catalog-boundary behavior.

– Protocol details named; not directly actionable by users `rust-v0.147.0`

`rust-v0.147.0-alpha.7` `rust-v0.147.0-alpha.2` `rust-v0.147.0-alpha.1`

28 **Secure inline visualization links in TUI** NEW 58

Adds secure, clickable inline visualization links rendered directly in the terminal UI; materializes viewer documents in a dedicated cache under `CODEX_HOME`, keyed by source and artifact thread IDs, so sandboxed sessions cannot overwrite viewer files before they are opened in a browser; disables visualization links for full-disk-write sessions and rejects viewer cache paths containing symbolic links, reusing materialized documents in memory when the source is unchanged.

– Names cache location; mechanism explained `rust-v0.148.0-alpha.12` `rust-v0.145.0`

`rust-v0.145.0-alpha.28`

29 **Configuration loading layers and overrides** BREAKING 56

Adds `packagedDefaults` as a new lowest-precedence configuration layer, with its source path reported through config diagnostics and the app-server protocol, returning an error when a configured packaged defaults file is missing; adds an override to skip project configuration at startup; removes config lockfile support.

– Names config layer; removal noted without detail `rust-v0.148.0-alpha.18`

`rust-v0.148.0-alpha.7`

30 **GPT-5.6 model catalog updates and field removals** BREAKING 54

Raises the GPT-5.6 maximum context window and updates bundled GPT-5.6 variants to 272,000-token context windows with refreshed model instructions, message configuration, reasoning-summary support, and personality instruction variables; adds a GPT-5.5 availability notice and exposes model upgrade retirement times in the model context; removes the `auto_review` and `permissions` message fields and the legacy `supports_reasoning_summary_parameter` flag from the bundled model catalog.

– Concrete numbers and removed fields named `rust-v0.148.0-alpha.22`

`rust-v0.148.0-alpha.14` `rust-v0.145.0-alpha.30` `rust-v0.145.0-alpha.27`

31 **`--psp` process-scoped PSP routing** NEW 53

Adds a `--psp` flag (made global, on both `app-server` and `exec-server`) to enable process-scoped PSP routing for first-party ChatGPT requests.

Start Codex with process-scoped PSP routing enabled for first-party ChatGPT requests.

— Simple flag, minimal elaboration beyond that rust-v0.148.0-alpha.6 rust-v0.147.0-alpha.13 rust-v0.147.0-alpha.10

32 **TUI session UX improvements** IMPROVED 53

Adds working-directory commands to the TUI letting users change the working directory from within the session; compacts successful command activity in the transcript for a cleaner view; enables editing queued messages with Vim history-up bindings; routes 'None of the above' selection in request input prompts to the notes editor on Enter in addition to Tab; supports drafting prompts while the TUI initializes, with resume and fork progress shown during startup; adds a running-task exit menu so Ctrl-C with an empty composer offers cancel-and-stay, exit-leaving-task-running, or stop-and-exit.

— Several UI tweaks named, no exact commands rust-v0.149.0-alpha.1 rust-v0.148.0 rust-v0.148.0-alpha.21 rust-v0.148.0-alpha.20 rust-v0.148.0-alpha.18 rust-v0.148.0-alpha.14

33 **Skill-level model delegation to Luna added then removed** DEPRECATED 52

Adds an optional `model` field to skill frontmatter (e.g. `model: luna`) and a `SkillModelDelegationInstruction` letting skills requesting Luna delegate to it when available, with bounded/validated model identifiers and instruction size limits; this skill model delegation support was removed later in the same window.

— Documents both addition and removal with named fields rust-v0.148.0-alpha.22 rust-v0.148.0-alpha.15

34 **Release artifact standardization and distribution** BREAKING 52

Standardizes Linux release artifacts to `codex-package-<target>` archives, dropping previously published redundant bundle archives; mirrors Rust release artifacts and metadata to Cloudflare R2 as an additional distribution channel; supports alpha hotfix release versions, mapping Python `aN.postM` to `-alpha.N.M` tags across installers, npm publishing, and release validation.

— Names artifact pattern and distribution channel rust-v0.147.0 rust-v0.147.0-alpha.13 rust-v0.145.0-alpha.29

35 **Token budget and context window controls** IMPROVED 52

Adds configurable goal token budget limits (`#37878`); supports model-owned token budget defaults, applying model-catalog-supplied budgets when no explicit token-budget configuration is set; treats `project_doc_max_bytes` as a single shared byte budget across all selected environments, truncating at the limit and skipping later environments once exhausted.

– Names config key and PR; mechanism thin `rust-v0.148.0-alpha.7` `rust-v0.148.0-alpha.4` `rust-v0.147.0-alpha.2` `rust-v0.147.0-alpha.1`

---

36 **MCP namespace description limit raised to 512 KiB** IMPROVED

50

Raises the MCP namespace tool-spec description limit from 1,000 bytes to 512 KiB, truncating only at a UTF-8 character boundary, so complete server instructions are preserved in tool-search source metadata instead of being hidden.

– Precise before/after numbers but no user action needed `rust-v0.147.0-alpha.6.6` `rust-v0.147.0-alpha.6.5` `rust-v0.147.0-alpha.6.3`

---

37 **Audio input and output support** NEW

50

Adds audio inputs and tool outputs supporting common local audio formats; adds audio variants in user input protocols, audio inputs forwarded to the Responses API, audio output in dynamic tools and code mode, and audio history gated by model input modalities; adds a `codex-utils-audio` workspace crate for canonicalizing audio inputs and estimating token usage.

– Names crate but mechanism thin overall `rust-v0.147.0-alpha.7` `rust-v0.145.0` `rust-v0.145.0-alpha.28`

---

38 **Unified exec enabled by default on Windows** IMPROVED

50

Enables `unified_exec` by default on Windows, exposing `exec_command` and `write_stdin` instead of `shell_command` on that platform.

– Names concrete tool substitutions on Windows `rust-v0.148.0-alpha.18`

---

39 **App-server experimental project and thread APIs** NEW

48

Adds experimental app-server project APIs and experimental thread queue APIs; supports pending environment attachment configuration so threads start immediately without blocking while attachment configuration is unavailable, with a new failed-attachment state that recovers on a ready update.

– Thin mention of experimental APIs, no specifics `rust-v0.149.0-alpha.1` `rust-v0.148.0-alpha.21` `rust-v0.148.0-alpha.20` `rust-v0.148.0-alpha.14`

---

40 **Enterprise and Business account plan support** IMPROVED

48

Adds Enterprise automation account plans and self-serve Business ProLite accounts; recognizes the `ent26` enterprise plan across authentication, account protocol, backend rate-limit payloads, and app-server schemas, treating it as an enterprise workspace plan for cloud-config eligibility and usage-limit guidance.

– Names plan identifier; little mechanism given `rust-v0.147.0-alpha.13` `rust-v0.147.0-alpha.2` `rust-v0.147.0-alpha.1` `rust-v0.146.0-alpha.8`

---

41 **Process diagnostics and resource metrics** NEW 48

Adds a `codex-diagnostics` crate that snapshots process ID, resident-memory measurements, and registered process-wide gauges, tracking live `CodexThread` instances via the `core.threads.live` gauge; adds general process diagnostics snapshots and exposes app-server and runtime activity diagnostics through the experimental API.

– Names crate and gauge; internal-facing `rust-v0.148.0-alpha.5` `rust-v0.148.0-alpha.4`

---

42 **Session and request metadata controls** NEW 48

Exposes the session ID to shell commands, enabling scripts and tools to reference the current session context; adds configurable Responses API request metadata for attaching custom metadata to API requests; preserves client-authored developer messages across context compaction when `retain_client_developer_messages` is enabled.

– Names config key; modest mechanism detail `rust-v0.148.0-alpha.14` `rust-v0.148.0-alpha.7`

---

43 **Esc key session interrupt and parallel stdin writes** NEW 47

Adds the Esc key as an interrupt mechanism to cancel a running agent task mid-session, now documented in the status indicator; runs `write_stdin` in parallel across unified-exec sessions, improving throughput when multiple shell sessions are active.

– Simple key mention, thin mechanism `rust-v0.145.0-alpha.30` `rust-v0.145.0-alpha.27`

---

44 **Durable per-thread message queue** NEW 46

Adds a command to queue messages for existing sessions so other processes can write to a running thread's durable queue, with idle threads dispatching them automatically; adds durable per-thread user submission queues.

– Mechanism described but no user command `rust-v0.148.0-alpha.14` `rust-v0.148.0-alpha.6` `rust-v0.147.0-alpha.10`

---

45 **Realtime V3 conversations and WebRTC sideband** NEW 44

Introduces streaming realtime V3 conversations, seeding sessions with initial text items and streaming realtime V3 Codex handoff output; routes WebRTC sideband joins to the Realtime API and reconnects WebRTC Realtime sideband transports automatically on disconnection; surfaces interactive requests in realtime conversations.

– Vague mechanism, no concrete config surface `rust-v0.148.0-alpha.23`

`rust-v0.148.0-alpha.22` `rust-v0.147.0-alpha.2` `rust-v0.145.0` `rust-v0.145.0-alpha.28`

---

46 **Managed in-app updates and feature gates** NEW 42

Adds managed in-app updates that let Codex CLI update itself automatically, with managed policy support and enterprise-plan recognition plus administrator controls for updates; adds managed gates for in-app chat and dictation.

– Vague on mechanism and enablement path `rust-v0.149.0-alpha.1` `rust-v0.148.0-alpha.22` `rust-v0.146.0-alpha.9.1` `rust-v0.147.0-alpha.1` `rust-v0.146.0`

---

47 **Workflow safeguards for directories, PRs and images** NEW 40

Prompts before trusting local project directories and rejects implicitly discovered bare Git repositories; adds Codex attribution links in pull request bodies; allows disabling the built-in image viewer.

– Three thin toggles bundled, no depth `rust-v0.148.0-alpha.6` `rust-v0.147.0-alpha.10`

---

48 **Cloud-managed sandbox profiles for `codex sandbox`** NEW 38

Loads cloud-managed profiles for `codex sandbox`, enabling centrally provisioned sandbox configurations.

– Thin one-line mention, no mechanism `rust-v0.147.0-alpha.2`

---

49 **Experimental thread config endpoint removed** BREAKING 35

Removes the experimental thread config endpoint from the app server.

– Named removal but no replacement given `rust-v0.148.0-alpha.22`

└──▷ BREAKING ON UPGRADE

! Rejects obsolete app-server permission profile fields — configs using removed fields will be refused on upgrade.

! Stops loading legacy managed config on Windows.

! The experimental thread config endpoint has been removed (Remove the experimental thread config endpoint).

! Skill model delegation support has been removed.

! Config lockfile support has been removed (Remove config lockfile support).

! The `--full-auto` flag on `codex exec` has been removed; use `--sandbox workspace-write` instead.

! Redundant Linux bundle archives are no longer published; use the standard `codex-package-<target>` release archives.

! Legacy `--full-auto` handling is removed from `codex exec` .

! Legacy Linux bundle archives are no longer published.

! Legacy collaboration mode variants have been removed (Remove legacy collaboration mode variants).

! The `auto_review` and `permissions` message fields and the legacy `supports_reasoning_summary_parameter` flag are removed from the bundled model catalog.

Anthropic Claude Code

15 RELEASES · 2026-07-20 → 2026-08-18

NOTES ↗ CODE ↗

Claude Code is an agentic coding tool that lives in your terminal, understands your codebase, and helps you code faster by executing routine tasks, explaining complex code, and handling git workflows - all through natural language commands.

> Claude Code shipped 15 releases (v2.1.216–v2.1.235) headlined by cross-machine session messaging (`SendMessage` / `ListAgents`), a new `claude self-hosted-runner` subcommand, sandbox credential masking, Claude Opus 5 as the default model, and a wave of plugin-marketplace, GitLab, Remote Control, and subagent-concurrency improvements alongside dozens of smaller CLI and VS Code refinements.

└──▷ WHAT SHIPPED · 72 FEATURES most completely described first ? what's the number?

01 **Sandbox credential masking** NEW 87

New sandbox masking options let commands read a sentinel value while a proxy substitutes the real secret on egress: `extract` and `onExtractNoMatch` for structured env values, `decode: "jwt"` with `maskClaims` for JWT-aware masking, `awsPairs` and `sigv4` for AWS SigV4 re-signing (require `network.tlsTerminate` , honored only from user/managed/ `--settings` settings), plus a new `mode: "mask"` for credential files on Linux and WSL serving a sentinel copy of the whole file or `extract` `-captured` spans, falling back to `deny` on macOS.

Mask a credential file in the sandbox so sandboxed commands see a sentinel value while the proxy substitutes the real secret on egress

```yaml
mode: "mask"
```

— Names all masking mechanisms with a config example v2.1.224 v2.1.221

02 **Cross-session messaging via SendMessage and ListAgents** NEW 83

Claude Code sessions can now message each other across machines: the new `SendMessage` API and `ListAgents` discovery (macOS and Linux) were introduced, then extended with `@ -mention` in the prompt to deliver directly to a named session, `/config` rows for 'Dialog expiry' and 'Messages from your other sessions' controlling accept/hold/refuse, and `crossSessionInbound` / `dialogExpiry` settings that hold messages to bypassed-permission sessions for approval. `SendMessage` now rejects oversized messages upfront, locks a confirmed Remote Control recipient by identity, can

initiate conversations with remote sessions by name, and is routed through the auto-mode permission classifier before dispatch; `ListAgents` reports when a session list was too long to check and labels sessions `offline` / `cloud`. Messages now render sender and body inline instead of a collapsed line.

— Names every setting and API surface across seven releases v2.1.235 v2.1.234 v2.1.232 v2.1.228 v2.1.225 v2.1.224 v2.1.222

---

03 **Remote Control connection reliability and client sync** IMPROVED

83

Remote Control gained auto-resume that continues a session automatically when a claude.ai usage limit resets (toggle in `/config` under 'Continue automatically at usage limit'), permission-mode and model sync to phones/Desktop/VS Code, and effort-level sync across connected clients. `claude rc` now applies the same enterprise-gateway availability check as interactive startup, `claude remote-control --continue` resumes the most recent session without an ID, reconnection persists for up to ~30 minutes across repeated blips, photos from the Claude app pass directly to Claude instead of a separate disk read, and attached web/mobile clients now see compaction progress, the post-compaction boundary, `/clear` resets, and a persistent connection-failure indicator with a reconnect shortcut instead of an 8-second toast.

Resume the most recent Remote Control session without looking up a session ID — useful when reconnecting after a network drop.

```
$ claude remote-control --continue
```

— Combines many Remote Control fixes with a runnable command v2.1.235 v2.1.234 v2.1.232 v2.1.229 v2.1.225 v2.1.224

---

04 **/code-review command improvements** IMPROVED

81

`/review` is now a proper alias of `/code-review`, which supports `/code-review <level> <pr#>` to review a specific PR, `/code-review ultra` for a deep cloud review, and now remembers the last effort level used. `/code-review` runs as a background subagent across effort levels to keep review output from flooding the conversation, and its empty-diff message for `ultra` now names the exact base ref and suggests passing an explicit one.

Run a deep cloud review on a specific PR number by passing an effort level and PR number to `/code-review`.

```
$ /code-review ultra 4321
```

— Named subcommand syntax with a runnable example v2.1.232 v2.1.223 v2.1.218

v2.1.216

05 **Subagent spawn limits and concurrency controls** BREAKING

81

New `CLAUDE_CODE_MAX_CONCURRENT_SUBAGENTS` environment variable (default 20) caps concurrently-running subagents. `CLAUDE_CODE_MAX_SUBAGENT_SPAWN_DEPTH` now controls nesting: subagents no longer spawn nested subagents by default, then max depth was raised to 3 (set the variable to `1` to disable nesting). The 200-subagent-per-session spawn cap was removed so long-running sessions aren't blocked from spawning new agents, though concurrency and depth limits still apply.

Limit how many background agents a single agentic session can fan out to, to control cost and resource usage.

```
$ CLAUDE_CODE_MAX_CONCURRENT_SUBAGENTS=5 claude
```

Allow subagents to spawn their own subagents up to a controlled depth when running multi-layer agentic pipelines.

```
$ CLAUDE_CODE_MAX_SUBAGENT_SPAWN_DEPTH=2 claude
```

Prevent deeply nested subagent spawning in CI pipelines where you want a flat, auditable agent graph.

```
$ CLAUDE_CODE_MAX_SUBAGENT_SPAWN_DEPTH=1 claude -p 'Run the full test suite and summarise failures'
```

— Two named env vars with runnable examples and defaults v2.1.217 v2.1.219 v2.1.224

06 **New plugin marketplace source types** NEW

78

Plugins can now come from three new source types: a local `command` source (e.g. from an IDE) that prints the plugin directory, is re-resolved each session, and applies without a restart (`mode: "link"` uses it in place); a new `archive` source that installs plugins from a zip over HTTPS without git or npm, with optional SHA-256 pinning; and GitLab marketplace support where bare `gitlab.com` repo URLs — including nested subgroups — clone the same way `github.com` URLs do.

Install a newly published GitLab-hosted plugin without manually refreshing the marketplace first.

```
$ /plugin install myplugin@gitlab.com/myorg/mygroup/my-marketplace
```

— Three named source types with an install example v2.1.232 v2.1.229 v2.1.224

07 **Marketplace allow/block policy controls** NEW 76

Managed settings gained `additionalMarketplaces` and `allowedMarketplaces` as friendlier aliases for `extraKnownMarketplaces` and `strictKnownMarketplaces`, a url-typed `blockedMarketplaces` entry that keeps blocking a bare repo URL even when classified as a git clone, and owner wildcard entries (`"owner/*"`) in `strictKnownMarketplaces` / `blockedMarketplaces` to allow or block an entire GitHub org in one entry.

Block all marketplace repos under a GitHub org in one managed-settings entry instead of listing each repo individually.

```json
{
  "blockedMarketplaces": ["my-org/*"]
}
```

— Named settings keys with a runnable config example v2.1.232 v2.1.223

08 **Plugin install and validation improvements** IMPROVED 73

`claude plugin validate` now checks a bare `.claude/skills` directory, reports `SKILL.md` frontmatter parse failures, accepts `'.'` as a valid `skills` path, and warns when a name would be rejected by Claude Desktop's managed marketplace sync. `/plugin install plugin@marketplace` now refreshes a stale marketplace catalog and retries before reporting not-found, and plugins now activate immediately when safe without requiring `/reload-plugins`.

— Names the exact command and files affected v2.1.232 v2.1.221

09 **Self-hosted runner support and performance** NEW 72

New `claude self-hosted-runner` subcommand turns your own machines or containers into execution environments for Claude Code web, mobile, and desktop sessions (Team and Enterprise plans). It gained server-supplied hook support matching managed environments, a faster session start (branch created without rewriting the working tree, eliminating two blocking server round trips), and now requires an explicit `--base-dir` for Windows startup since there is no default checkout directory on Windows.

— Named subcommand and flag, mechanism explained v2.1.224 v2.1.233 v2.1.229

10 **Vertex AI and Bedrock reliability improvements** IMPROVED 72

New `ANTHROPIC_BEDROCK_REGION_PREFIX` environment variable lets Bedrock users prefer a specific cross-region inference profile instead of the `AWS_REGION` `-derived` default. Expired/missing Google Cloud credentials for Vertex AI now fail within seconds instead of minutes, tool search was re-enabled on Vertex AI for Claude 4.5-generation and newer models, and gateway streaming responses now send SSE keepalive pings during long thinking pauses to prevent idle-timeout disconnects on Vertex and Bedrock upstreams.

Route Bedrock inference to a specific cross-region profile rather than the one inferred from `AWS_REGION`.

```
$ export ANTHROPIC_BEDROCK_REGION_PREFIX=us-east
claude
```

— Named env var with example plus three provider fixes `v2.1.224` `v2.1.228` `v2.1.221` `v2.1.229`

11 **Gateway settings approval dialog and sandbox.ripgrep lockdown** 71

BREAKING

`sandbox.ripgrep` is now honored only from user, managed, and `--settings` settings — project settings can no longer override it. The managed-settings approval dialog now shows endpoint URLs and requires approval for server-managed sandbox binary overrides (`sandbox.bwrapPath`, `sandbox.socatPath`, `sandbox.ripgrep`), while benign feature and cost toggles no longer trigger that approval prompt.

— Names exact keys but no example config `v2.1.232` `v2.1.218`

12 **Gateway boot validation hardening** BREAKING 70

The Gateway's `desktop:` overlay now accepts every released Desktop setting (previously 11 hand-listed keys) and is validated at boot against Desktop's schema, with unknown/invalid keys now failing boot. Empty `managed.policies[].match.groups` / `admin.admin_groups` entries and malformed `email_domain` values also now fail at boot instead of silently matching no one or granting admin access.

— Detailed before/after but requires gateway ops context `v2.1.232`

13 **VSCode UI improvements** NEW 70

VS Code gained session groups in the sidebar (right-click to create, rename, or delete; Cmd/Ctrl- or Shift-click to move several at once), a resizable `/btw` side-question panel, and a Focus view (`Ctrl+Alt+F` or the 'Claude Code: Toggle Focus view' command) that hides tool activity behind an expandable per-turn summary with a live running-tool indicator.

— Names exact UI paths and keybinding `v2.1.229` `v2.1.221`

14 **Unrecognized model ID handling** BREAKING 68

Print mode now writes a `[claude-code:unrecognized_model]` line to stderr for unrecognized model IDs (silence with `modelOverrides`). Sessions on unrecognized model IDs are now enforced against the assumed context window via auto-compaction by default; set `CLAUDE_CODE_DISABLE_UNKNOWN_MODEL_WINDOW_ENFORCEMENT=1` to restore prior behavior.

— Names stderr marker, config key, and opt-out env var `v2.1.233` `v2.1.223`

15 **Todo/task-tracking tools disabled by default on newer models**

68

BREAKING

`TaskCreate` , `TaskGet` , `TaskUpdate` , `TaskList` , and `TodoWrite` are no longer available by default on Opus 4.8, Sonnet 5, Fable 5, Mythos 5, and newer models; set `CLAUDE_CODE_ENABLE_TODO_TOOLS=1` to re-enable them.

Re-enable todo/task-tracking tools on a newer model (e.g. Sonnet 5) where they are now off by default.

```
$ CLAUDE_CODE_ENABLE_TODO_TOOLS=1 claude
```

— Named tools and env var with runnable example `v2.1.233`

16 **AWS gateway reference deployment assets** NEW 68

New AWS reference deployment assets under `examples/gateway/aws/` — including `setup.sh` , `Dockerfile` , `gateway.yaml.example` (Bedrock upstream, Okta IdP), and a `terraform/` module — for running the Claude apps gateway on AWS with Amazon Bedrock.

— Names every asset file in the deployment `v2.1.217`

17 **CLAUDE_CODE_TOOL_MEMORY_LIMIT env var** NEW 68

New `CLAUDE_CODE_TOOL_MEMORY_LIMIT` environment variable (opt-in, Linux only) caps memory for Bash tool commands via cgroups, preventing a runaway build from stalling the session.

Cap memory for a Bash tool invocation on Linux so a runaway build step cannot exhaust system memory and stall your session.

```
$ CLAUDE_CODE_TOOL_MEMORY_LIMIT=2g claude
```

— Named env var with mechanism and runnable example v2.1.233

18 **Skill frontmatter and execution defaults** IMPROVED 66

Skill and plugin frontmatter now accepts `yes` / `no` / `on` / `off` / `1` / `0` (case-insensitive) as boolean values alongside `true` / `false`. Skills with `context: fork` now run in the background by default; opt out per-skill with `background: false`.

Opt a specific skill out of the new default background execution when it uses `context: fork`.

```yaml
context: fork
background: false
```

— Named frontmatter keys with a config example v2.1.218

19 **GitLab token redaction and glab credential protection** NEW 66

Adds secret redaction for GitLab token families `glrt-`, `gloas-`, `glptt-`, `glagent-`, `glimt-`, `glsoat-`, `glcbt-`, `glft-`, `glffct-`, plus full redaction of routable `glpat-` / `gldt-` tokens; the `glab` CLI config store now receives the same sandbox and credential-path protection as `gh`.

— Enumerates every token prefix redacted v2.1.232

20 **/goal reliability improvements** IMPROVED 66

New `CLAUDE_CODE_GOAL_CHECKIN_MINUTES` environment variable (set to `0` to opt out) makes Claude check in when background tasks keep a `/goal` waiting 30+ minutes instead of indefinitely; `/goal` also now clears itself with a notice when a turn dies on an unrecoverable error (revoked auth, exhausted credit balance, or context overflow) instead of staying armed.

Tune how long a `/goal` waits before Claude checks in on stalled background tasks — set to 0 to disable check-ins entirely.

```
$ CLAUDE_CODE_GOAL_CHECKIN_MINUTES=15 claude
```

— Named env var and default behavior with example `v2.1.234`

---

21 **sandbox.network.strictAllowlist deny-without-prompt** NEW

64

New `sandbox.network.strictAllowlist` setting denies non-allowlisted hosts for sandboxed commands outright, with no user prompt.

Lock sandboxed commands to only the hosts you explicitly allow — no prompt, just deny everything else.

```json
{
  "sandbox": {
    "network": {
      "strictAllowlist": true
    }
  }
}
```

— Single config key with runnable example `v2.1.219`

---

22 **sandbox.filesystem.disabled decouples isolation** NEW

64

New `sandbox.filesystem.disabled` setting skips filesystem isolation for sandboxed commands while keeping network egress control active.

Allow a sandboxed agent to access the filesystem freely while still restricting outbound network traffic.

```yaml
sandbox.filesystem.disabled: true
```

— Named config key with example, thin scope description `v2.1.216`

---

23 **Claude Opus 5 release and fast mode changes** BREAKING

64

Adds Claude Opus 5 (`claude-opus-5`) as the new default Opus model, with a 1M context window and fast mode at $10/$50 per Mtok. Opus 4.7 is removed from fast mode, so `/fast` now applies only to Opus 5 and Opus 4.8, and an announcement now appears when fast mode changes due to a model switch via `/config model=<x>` or Remote Control.

— Names model ID, context size and pricing `v2.1.219` `v2.1.218`

24 **Workflow fan-out prefix staggering** IMPROVED 64

Workflow fan-outs now stagger same-prefix sibling agents so subsequent agents read the cached prompt prefix instead of re-paying it; set `CLAUDE_CODE_WORKFLOW_PREFIX_STAGGER_MS=0` to disable it.

Disable workflow fan-out staggering in environments where prompt-prefix caching is not needed or causes timing issues.

```
$ CLAUDE_CODE_WORKFLOW_PREFIX_STAGGER_MS=0 claude
```

— Named env var with runnable disable example `v2.1.229`

25 **workflowSizeGuideline and Dynamic workflow default size** NEW 64

New `workflowSizeGuideline` settings key lets the Dynamic workflow size guideline be set from any settings file (hiding the `/config` row while active); Dynamic workflows now default to a medium size guideline of fewer than 15 agents, changeable via Dynamic workflow size in `/config`.

Pin the Dynamic workflow size guideline in a shared settings file so all team members get the same agent-count target without needing to touch `/config`.

```json
{
  "workflowSizeGuideline": "medium"
}
```

— Named config key with example and numeric default `v2.1.219`

26 **--teleport flag for continuing cloud sessions locally** NEW 64

New `--teleport <session id>` flag on `claude`, surfaced as a `/teleport` hint inside cloud sessions, lets practitioners seamlessly continue a cloud session locally.

Pick up a cloud session locally after starting work in the browser — use the teleport hint printed in the cloud session.

```
$ claude --teleport <session id>
```

— Named flag with a runnable example `v2.1.223`

27 **GitLab merge request integration** NEW 63

Repos with a GitLab remote and an authenticated `glab` CLI now show MR state: a footer/statusline badge displays `!N` with draft/pending/green states, and GitLab MR URLs are supported by the `--worktree` flag and the `claude agents` view.

— Names the flag and view but no runnable example v2.1.234 v2.1.233

28 **MCP connection diagnostics** IMPROVED 63

`claude mcp list` and `/mcp` now show HTTP status codes and error text when a server fails to connect, plus a warning for MCP config values with hidden leading or trailing whitespace. The headless stream-json init event gained an `mcp_server_errors` field listing `--mcp-config` entries skipped by config validation, with terminal runs printing a startup warning.

— Names commands and the new stream-json field v2.1.219 v2.1.218

29 **CLAUDE_CODE_PROJECT_DIR_NAME env var** NEW 62

New optional `CLAUDE_CODE_PROJECT_DIR_NAME` environment variable lets hosts that give each session its own config directory choose a short name for the per-project transcript directory.

Give a multi-user host a predictable, short transcript directory name per project instead of a generated one.

```
$ CLAUDE_CODE_PROJECT_DIR_NAME=my-api-service claude
```

— Named env var with runnable example v2.1.234

30 **Mid-turn command availability** IMPROVED 62

`/permissions` can now be opened while Claude is actively working, with rule changes applying immediately to the rest of the current turn; `/add-dir <path>` can now be invoked mid-turn, and `/add-dir`, `/autocompact`, `/theme`, `/help`, `/config`, and `/advisor` dialogs all now open mid-turn in the fullscreen TUI.

— Lists every dialog affected v2.1.234

31 **CLAUDE_CODE_WEBFETCH_CACHE_TTL_MS env var** NEW 62

New `CLAUDE_CODE_WEBFETCH_CACHE_TTL_MS` environment variable overrides the WebFetch session URL cache TTL (default: 15 minutes).

Shorten the WebFetch URL cache TTL to 5 minutes when you need Claude to re-fetch frequently updated pages within a session.

```
$ CLAUDE_CODE_WEBFETCH_CACHE_TTL_MS=300000 claude
```

— Named env var with default and example v2.1.233

32 **Background session lifecycle improvements** IMPROVED 61

Background sessions now commit and push to preserve work, open a draft PR only when the task calls for one, follow `CLAUDE.md` git instructions, and always report where work lives; `/status` now shows the session kind (`interactive` , or a background job that is `attached` or `unattended`); and `/mcp` / `/install-github-app` now park a 'needs input' request in the agent view when no client is attached.

— Names the status values and affected commands v2.1.221 v2.1.216

THINNER COVERAGE BELOW

33 **Remote Control auto-start restricted to user-scope settings** 58

BREAKING

Remote Control auto-start can no longer be enabled via repo-local `.claude/settings.json` or `.claude/settings.local.json` ; it must now be enabled at user scope via `/config` (those files can still turn it off).

— Clear before/after but no example v2.1.222

34 **Nested subagent text forwarding in stream-json** NEW 57

Subagents spawned at depth-2+ now appear in stream-json output when `--forward-subagent-text` is set, keyed by their spawning Agent `tool_use` id.

— Names the flag, no example command v2.1.219

35 **Auto mode permission classifier expansion** IMPROVED 56

Auto mode now routes more decisions through its permission classifier instead of opening dialogs: dangerous-rm, background- `&` , and suspicious-Windows-path checks; Bash commands the static analyzer can't prove read-only in plan mode; and messages sent via `SendMessage` , now checked before dispatch. Auto-mode permission checks for parallel tool calls are also now cache-efficient, reusing the cached conversation prefix across decisions.

— Names the checks affected but no config lever v2.1.222 v2.1.218 v2.1.221

36 **claude-api skill improvements** IMPROVED 55

The built-in `claude-api` skill's context cost was cut from ~200k+ tokens to ~25k by loading reference docs on demand, and it gained a `prompt-audit` subcommand for auditing prompts and tool descriptions for patterns written for older models.

— Concrete token numbers and named subcommand v2.1.234 v2.1.221

37 **Emoji autocomplete shortcodes and setting** IMPROVED 55

New `emojiCompletionEnabled` setting controls emoji shortcode autocomplete in the prompt input (type `:heart:` to insert or `:hea` for suggestions), and autocomplete now also accepts common alternate shortcodes like `:thumbsup:`, `:thumbsdown:`, and `:love:`.

— Named setting and shortcode examples v2.1.221 v2.1.217

---

38 **Sandbox and Bash permission hardening** IMPROVED 54

IPv6 literals in sandbox network domain lists are now bracketed (e.g. `[::1]:443`) and ambiguous spellings are enforced fail-closed and flagged by `/doctor`. Bash input redirections (`< file`) are now permission-checked on all platforms, matching existing argument-spelling behavior.

— Two thin hardening notes with named syntax v2.1.232 v2.1.229

---

39 **FORCE_HYPERLINK env var for footer PR badge** NEW 54

New `FORCE_HYPERLINK=0` environment variable opts out of footer PR badge links being rendered as clickable hyperlinks even when terminal support cannot be detected (e.g. over ssh/tmux).

— Named env var, no runnable example given v2.1.217

---

40 **DirectoryAdded hook** NEW 53

New `DirectoryAdded` hook fires after `/add-dir` or the SDK `register_repo_root` control request registers a new working directory mid-session.

— Names the hook and triggering calls, no example v2.1.219

---

41 **Hardened skills synced from claude.ai** IMPROVED 52

Skills synced from claude.ai no longer shadow local commands or MCP prompts, their descriptions are sanitized and labeled, and skill bodies can no longer run `!` commands or expand `@` files on your machine.

— Describes mechanism but no config surface to act on v2.1.228

---

42 **/ultrareview improvements** IMPROVED 52

`/ultrareview` now runs a branch review with descriptive arguments like 'review my auth changes' applied as a note to findings, with invalid arguments surfacing actionable error feedback; its diff-too-large error now shows the configured limits, measured diff size, and the largest contributing files.

— Describes behavior, no exact syntax given v2.1.218 v2.1.216

---

43 **Optional spellcheck setting for prompt input** NEW 52

New optional `spellcheck` setting underlines misspelled words in the prompt input as you type, backed by your installed `aspell`, `hunspell`, or `ispell`.

– Named setting and backing tools, no exact key path v2.1.235

44 **/fork command improvements** IMPROVED 52

`/fork` now creates a new worktree for the forked session instead of sharing the original session's checkout, and its confirmation now shows the new session name, `claude attach` id, and a note when the copy shares your checkout — all on one line.

– Names the command and confirmation contents v2.1.221 v2.1.216

45 **forward_user_identity setting for apps gateway** NEW 52

New `forward_user_identity` opt-in setting on Anthropic upstreams in the apps gateway sends the signed-in user's identity as headers so a downstream proxy can attribute spend per user.

– Named setting, no example config shown v2.1.233

46 **Spend limit messaging improvements** IMPROVED 50

The usage warning now supports gateway spend limits: when a cap is reached it names the limit, its reset time, and the operator's message (requires gateway v2.1.225). The spend-limit adjustment prompt also now shows the server's reason when a requested change is rejected.

– Describes messaging content, no direct action v2.1.225 v2.1.216

47 **Accessibility (screen reader) improvements** IMPROVED 50

In `--ax-screen-reader` mode, the `/effort` selector now renders as a numbered list with a typed-number prompt and hint/dialog text is no longer clipped, and deleted text is now announced for Option+Delete, Ctrl+W, Cmd+Backspace, Ctrl+U, and Ctrl+K.

– Names key combos, no way to enable beyond mode v2.1.233 v2.1.218

48 **/commit-push-pr dangerous flags no longer auto-approved** BREAKING 48

`/commit-push-pr` no longer auto-approves git/gh commands carrying dangerous flags such as `--force`, `--amend`, and `--no-verify`.

– Names exact flags affected, thin scope v2.1.229

49 **subagent_type "fork" default** IMPROVED 48

`subagent_type: "fork"` subagents are now enabled by default, so forked subagents inherit the full conversation and prompt cache; non-teammate agent spawns in interactive sessions now run in the background by default.

– Names the type value, no config lever given v2.1.232

50 **Diff rendering uses raw git blob content** IMPROVED 48

The `/diff` view, Remote Control workspace diff, and file-edit diffs in Claude Code web sessions now use raw git blob content, bypassing workspace-configured diff drivers and textconv for cleaner output.

— Clear mechanism, no toggle to control it `v2.1.222`

51 **Worktree isolation blocks destructive git commands** IMPROVED 47

Worktree-isolated sessions and their subagents can no longer run destructive git commands against the main checkout; isolation now covers file edits and Bash across every session type.

— Clear scope, no config surface named `v2.1.222`

52 **/context and /rewind safety messaging** IMPROVED 47

`/context` now shows an explicit warning when the conversation exceeds the context window, with a failed `/compact` displaying as an error; `/rewind` now reports how many tracked paths it skipped rather than silently restoring or deleting files through symlinks or hard links.

— Two named commands with clear before/after `v2.1.216`

53 **selection:clear keybinding action** NEW 46

New `selection:clear` keybinding action can be bound to a key to clear an in-app text selection; also works in the agents view.

— Named action, no default binding given `v2.1.234`

54 **Feedback-survey transcript share includes model settings** NEW 46

Feedback-survey transcript share now also uploads (with consent) the last request's model settings — system prompt (including `CLAUDE.md`), tool definitions, and model parameters — with secrets redacted.

— Names what's uploaded, consent-gated feature `v2.1.224`

55 **Workspace trust dialog improvements** IMPROVED 43

`claude agents` now shows a workspace trust prompt for untrusted directories, matching the existing `claude` command behavior, and trust dialogs now name the specific repository root the grant covers.

— Names affected command, thin mechanism `v2.1.225` `v2.1.218`

56 **Context-limit error indicates auto-compact off** IMPROVED 42

The context-limit error now indicates when auto-compact is off and directs users to `/config` to re-enable it.

— Clear navigation path in /config v2.1.235

57 **Removes 'Default teammate model' setting** DEPRECATED 41

Removes the 'Default teammate model' setting from `/config`; agent-team teammates now use the leader's model unless the spawn command names one explicitly.

— Named setting removed, clear fallback behavior v2.1.234

58 **Reject agent names containing ':'** BREAKING 41

Agent markdown files now reject agent names containing `:`, reserving that character for plugin namespacing; existing agents with `:` in their name will be refused on load.

— Clear breaking rule with the exact character v2.1.218

59 **Session naming improvements** IMPROVED 40

Auto-generated session titles are now short and specific (e.g. 'Login button bug') rather than sentences restating the request, and interactive sessions on the same machine now enforce unique names — a collision produces a `name-word-word` variant automatically.

— Cosmetic change with example naming pattern v2.1.234 v2.1.232

60 **Write tool allows newer models to overwrite unread files** IMPROVED 40

The Write tool now allows newer models to overwrite an existing file they haven't read this session, matching the Edit tool's rules; older models still require a read first.

— Clear rule change, model-dependent scope v2.1.228

61 **Windows startup no longer spawns powershell.exe** IMPROVED 38

Windows startup no longer spawns `powershell.exe` to read process creation times, eliminating prompts from endpoint security tools that gate `powershell.exe`.

— Single-line fix, no action needed by reader v2.1.221

62 **Subagent model restriction warning** NEW 38

Adds a warning when workflow agents, forked skills, slash commands, or resumed background agents have their requested subagent model restricted and the parent model is substituted instead.

— Describes the warning, no way to configure it v2.1.223

63 **Transcript write failure warnings** IMPROVED 38

Claude Code now warns when transcript writes are failing (e.g. disk full) or when session saving is off due to an inherited environment variable, instead of silently losing transcripts.

— Clear scenario, no config key named v2.1.217

64 **Stats panel cache token breakdown** IMPROVED 36

Stats panel now counts cache tokens in its token totals, with a breakdown by input, output, cache read, and cache write.

— Named metrics but no interaction needed v2.1.221

65 **/login OAuth token warning repeats after login** IMPROVED 35

`/login` now repeats the `CLAUDE_CODE_OAUTH_TOKEN` override warning after a successful login.

— Single-line fix naming the env var v2.1.229

66 **Fullscreen scrollback retains full pre-compaction history** IMPROVED 33

Fullscreen mode now retains the full pre-compaction history in scrollback across repeated compactions instead of only the most recent interval.

— Clear before/after, no config to act on v2.1.224

67 **claude setup-token rejects extra arguments** IMPROVED 32

`claude setup-token` now rejects unexpected extra arguments instead of silently ignoring them.

— One-line CLI validation fix v2.1.234

68 **/deep-research manual invocation only** BREAKING 30

`/deep-research` now starts only when invoked manually — Claude no longer launches it autonomously.

— One-line behavior change, minimal detail v2.1.218

69 **Prompt markdown rendering in transcript** IMPROVED 29

Your own prompts in the transcript now render markdown — highlighted code blocks, inline code, lists — the same way replies do.

— Cosmetic change, no action needed v2.1.234

70 **disable-model-invocation refusal asks user to run skill** IMPROVED

29

The `disable-model-invocation` refusal now instructs Claude to ask the user to run the skill instead of attempting to replicate its workflow.

— Single-line behavior change `v2.1.222`

71 **Compaction progress diagnostics** IMPROVED 27

Compaction progress now shows a retry countdown and stall hint, not just a progress bar.

— Minimal UI detail, no config `v2.1.228`

72 **Removes ultraplan feature** DEPRECATED 21

The ultraplan feature has been removed.

— No detail beyond removal notice `v2.1.222`

└──▷ BREAKING ON UPGRADE

! Todo/task-tracking tools (`TaskCreate`, `TaskGet`, `TaskUpdate`, `TaskList`, `TodoWrite`) are no longer available by default on Opus 4.8, Sonnet 5, Fable 5, Mythos 5, and newer models; set `CLAUDE_CODE_ENABLE_TODO_TOOLS=1` to re-enable them.

! Gateway: unknown or invalid keys in the `desktop:` overlay now fail boot (previously accepted silently).

! Gateway: empty `managed.policies[].match.groups` / `admin.admin_groups` entries and malformed `email_domain` values now fail at boot (previously matched silently or granted admin access).

! `sandbox.ripgrep` set in project settings is no longer honored; it must be in user, managed, or `--settings` settings.

! Self-hosted runner Windows startup now requires an explicit `--base-dir`; there is no default checkout directory on Windows, so existing Windows runner configs without `--base-dir` will fail to start.

! Remote Control auto-start can no longer be enabled by repo-local settings (`.claude/settings.json` or `.claude/settings.local.json`); repos relying on those files to turn Remote Control on must switch to user-scope configuration via `/config`.

! Opus 4.7 is removed from fast mode; `/fast` now applies only to Opus 5 and Opus 4.8 — workflows pinned to Opus 4.7 in fast mode will no longer run in fast mode.

! Subagent spawn depth default raised from 1 to 3; pipelines that relied on nesting being blocked at depth 2 will now allow deeper nesting unless `CLAUDE_CODE_MAX_SUBAGENT_SPAWN_DEPTH=1` is set.

! Agent markdown files now reject agent names containing `:` — any existing agent whose name includes `:` will be refused on load.

! Subagents no longer spawn nested subagents by default; set `CLAUDE_CODE_MAX_SUBAGENT_SPAWN_DEPTH` to restore deeper nesting.

WAS THIS USEFUL? 👍 👎

Cline

40 RELEASES · 2026-07-22 → 2026-08-19 NOTES ↗ CODE ↗

Autonomous coding agent as an SDK, IDE extension, or CLI assistant.

> Cline shipped a new native macOS desktop app with auto-updates, then rapidly layered on cloud agent sessions, voice input, and inline image generation; in parallel, the CLI, SDK, and desktop clients gained OAuth-secured MCP server management, mid-task web search, hard-blocked file edits in Plan mode, free-tier `cline-free` models, and broadened Vertex AI, Claude Opus 5, and other provider support.

└──▷ WHAT SHIPPED · 63 FEATURES most completely described first ? what's the number?

01 MCP OAuth authentication, uninstall command, and timeout handling

NEW 98

Remote SSE MCP servers surface an OAuth authorization prompt on a 401 response instead of failing, supporting pre-registered OAuth clients (client ID/secret) where dynamic registration is unavailable, with stored tokens invalidated on config change. Settings → MCP lets you authorize a server, view auth status, and cancel or retry authorization. New `cline mcp uninstall <name>` (alias `cline mcp rm <name>`) removes a named server from `cline_mcp_settings.json`. The per-server `timeout` field in `cline_mcp_settings.json` is honored for `initialize`, `tools/list`, and `tools/call`, replacing hardcoded 1.5s/5s limits, defaulting to 60 seconds and clamped to 1–3600 seconds. A hung MCP server no longer blocks session creation; unconfigured stdio servers get a 30-second initialize budget. MCP tool results render as readable text in the TUI instead of escaped JSON, MCP errors surface on the individual server row instead of page-level, and server cards are consistent across marketplace views with a single uninstall action and setup guidance.

Remove an MCP server you no longer need from the CLI configuration.

```
$ cline mcp uninstall
```

Remove a named MCP server from your settings without manually editing JSON.

```
$ cline mcp uninstall docs
# or using the alias:
cline mcp rm docs
```

Allow a slow MCP server up to 30 seconds to initialize or respond to tool calls, preventing spurious failures.

```json
{
  "mcpServers": {
    "my-slow-server": {
      "command": "npx",
      "args": ["-y", "my-mcp-server"],
      "timeout": 30
    }
  }
}
```

Set a custom MCP server timeout so slow servers don't hit the default 60-second limit during initialization and tool calls.

```json
{
  "mcpServers": {
    "my-slow-server": {
      "command": "node",
      "args": ["server.js"],
      "timeout": 120
    }
  }
}
```

— Names exact command, config key, defaults, and clamp values. `desktop-v0.0.11` `v4.1.7` `cli-v3.0.52` `sdk/sdk/v0.0.72` `desktop-v0.0.10` `desktop-v0.0.8` `cli-v3.0.48` `sdk/sdk/v0.0.67` `desktop-v0.0.5`

---

02 **CLI theme support** NEW 90

Adds `CLINE_THEME` environment variable to override the persisted theme choice at startup, and a `/theme` command plus Theme row in `/settings` to select from 11 built-in themes (Auto, Cline Dark, Cline Light, Tokyo Night, Gruvbox Dark, Nord, Dracula, Catppuccin Mocha, One Dark, Solarized Dark, Solarized Light) with live preview in the interactive TUI.

Set a persistent dark theme at startup without opening the TUI settings — useful in CI or when launching from a script.

```
$ CLINE_THEME='Tokyo Night' cline
```

Switch themes interactively during a session to match your terminal palette.

```
$ /theme
```

— Exact env var, command, and full theme list — runnable now. `cli-v3.0.50`

---

03 **Persistent CLI settings via GlobalSettingsSchema** NEW

80

Plan/act mode, tool auto-approve, and compaction mode are now persisted across CLI restarts in global settings via `GlobalSettingsSchema` fields `planActMode`, `toolAutoApprove`, and `compactionEnabled`, with cross-process-safe writes so two hosts no longer clobber each other's changes.

— Exact schema field names given; settings persist without user action. `cli-v3.0.47` `sdk/sdk/v0.0.66`

---

04 **Headless schedule auto-approve and mode selection** NEW

80

Adds `--mode <act|plan|yolo>` to `cline schedule create` and `cline schedule update`, with `yolo` now the default mode so headless routines run unattended without approval prompts; headless scheduled routines default to auto-approve and no longer prompt for interactive input.

Create a headless one-time routine that runs without approval prompts, using the new explicit yolo mode flag.

```
$ cline schedule create 'Nightly dependency audit' --cron '0 2 * * *' --prompt 'Check for outdated npm dependencies and open a PR with updates' --workspace /path/to/repo --mode yolo
```

— Exact flag and a runnable schedule-create command are given. `sdk/sdk/v0.0.66` `desktop-v0.0.4`

---

05 **Plan mode hard-blocks file-editing shell commands** NEW

80

Plan mode now hard-blocks file-editing shell commands at the tool level via `run_commands` instead of relying on prompting alone: file-manipulation commands, in-place editors (`sed -i`, `perl -i`), output redirection, mutating git subcommands, package installs, and nested command strings (`sh -c`, `eval`, `sudo`, `xargs`) are rejected with a tool error on Linux, Windows, and PowerShell, while read-only investigation commands still run.

— Enumerates exact blocked command patterns and the tool involved. v4.1.4

desktop-v0.0.9 cli-v3.0.50 sdk/sdk/v0.0.70

06 **Web search during tasks** NEW 80

Enables web search during a task for models that support it, toggled via the 'Web Search' setting (the `web_search` model tool, off by default) in Settings/Feature Settings; search calls and results appear in the transcript/conversation and persist across session reopens and reloads. The Web Search toggle now shows which connected providers support built-in web search and warns with a link to Models when none do.

Enable live web search for a model that supports it so the agent can look up current information during a task.

> 📍 Enable the `web_search` model tool in Cline settings, then run: cline "What are the CVEs disclosed in the last 24 hours affecting OpenSSL?"

— Names the `web_search` tool with a runnable example query. desktop-v0.0.14

desktop-v0.0.13 v4.1.10 sdk/sdk/v0.0.75

07 **`getShellInvocation()` replaces `getShellArgs()`** DEPRECATED 80

Adds getShellInvocation() to replace the now-deprecated getShellArgs(), routing PowerShell commands over UTF-8 stdin instead of the command line — fixing non-ASCII command corruption and lifting the Windows command-line length cap; stdin write failures now surface as a command error instead of hanging.

— Names both functions and the exact migration path for callers. sdk/sdk/v0.0.68

08 **Cloud agent sessions (preview)** NEW 80

Adds cloud sessions (preview) — connect GitHub, pick a repository and branch, and run sessions in Cline's cloud with transcripts, approvals, and queued prompts synced across devices; enable via the Cloud sessions toggle in Settings. Supports renaming cloud sessions and switching models mid-session, gated behind an explicit settings toggle with a GitHub connect onboarding panel.

Enable cloud sessions to hand off a coding session between devices without losing context.

> 📍 In the desktop app, go to Settings and enable the 'Cloud sessions' toggle. During onboarding, connect your GitHub account, then pick a repository and branch to start a cloud session.

— Exact settings toggle and onboarding steps to enable it. desktop-beta

desktop-v0.0.14-beta.1

09 **Desktop app performance overhaul** IMPROVED 75

Streaming responses now coalesce token updates instead of re-rendering the entire chat per token, eliminating stutter during long generations. The animated background renders at a locked 60 fps (up from ~10 fps) and composer keystroke handling drops from 245 slow keystrokes to 3. App boot fetches the provider catalog once instead of three times, and the native folder picker and command execution no longer block the UI while the sidecar writes session logs or discovers the editor.

— Concrete before/after numbers given but nothing for reader to invoke. desktop-v0.0.5

---

10 **Vertex AI Fable 5 support and full regional model catalog** NEW 75

Adds Claude Fable 5 (`claude-fable-5`) to the Vertex AI model catalog (pricing shown as unknown since Vertex bills region-dependently); supports entering any Vertex model ID by hand, including models not yet in the catalog, with custom Vertex model IDs passed through unchanged so Claude-style IDs route to the Anthropic-on-Vertex path. Shows the full model catalog for every Vertex region instead of a hardcoded list of global-endpoint models; the global-region picker no longer hides catalog models, and unsupported region/model combinations now fail at request time with recovery guidance.

— Names the exact model ID and describes the picker change. desktop-v0.0.12 cli-v3.0.53 sdk/sdk/v0.0.73 v4.1.8

---

11 **Compaction honors configurable max output tokens** IMPROVED 70

Compaction/summarization now respects an explicit `max-output-tokens` setting (also surfaced as the 'Max Output Tokens' setting in the desktop app), defaulting to 4096 and lowered when the model reports a lower cap, instead of a hardcoded 1024-token cap — unblocking reasoning models that previously exhausted the budget thinking with no summary produced; a diagnostic is now logged when a summary comes back empty.

— Names the setting and default value, though not a direct flag. desktop-v0.0.12 sdk/sdk/v0.0.73

---

12 **Session forking APIs for editing earlier prompts** NEW 70

Adds session forking and user-run message APIs so a host can edit an earlier prompt: fork the session before a selected user run, trim checkpoint history, and restore prior messages. Adds `ClineCore.readLiveMessages` to read a resident session's in-memory transcript, so a plan/act rebuild during an in-flight turn starts from the full history instead of an empty one.

Read the live in-memory transcript of an in-flight session to preserve history before a plan/act rebuild.

```typescript
import { ClineCore } from "@cline/sdk"

const messages = await ClineCore.readLiveMessages(sessionId)
```

— Names exact API and includes a runnable code snippet. sdk/sdk/v0.0.67

13 **Connector reliability across messaging platforms** IMPROVED

65

Connector threads across Slack, Discord, Telegram, Linear, Google Chat, and WhatsApp now automatically recover when their bound session is gone, dropping the stale binding and replaying the turn against a new session; connector sessions persist and automatically reconnect after a daemon or hub restart. Connector launch collisions are eliminated — an instance is claimed before socket mode opens, the hub supervises connector processes, `doctor` / `connect` skip connectors already starting, connector tools are enabled by default, and Slack's greeting is no longer replayed on reconnect. Telegram slash commands (e.g. `/clear`) are now delivered to the connector command host instead of being silently dropped.

— Lists exact platforms and commands but recovery is automatic. cli-v3.0.50 cli-v3.0.48 sdk/sdk/v0.0.66

14 **Provider compatibility fixes for Claude, Bedrock, and OpenAI-compatible endpoints**

IMPROVED

65

Claude 4.6+ and 5.x (adaptive-era) models are now accepted without the '`thinking.type.enabled` is not supported' rejection — the model catalog carries reasoning metadata and infers it for unlisted or hand-typed adaptive IDs. Bedrock prompt caching now correctly records cache reads and writes using Converse `cachePoint` markers, and Bedrock foundation models route through geo inference profiles. Reasoning models on OpenAI-compatible endpoints now receive `max_completion_tokens` instead of the previously rejected `max_tokens`.

— Names exact wire fields fixed; behavior corrects itself automatically. desktop-v0.0.9 cli-v3.0.50 sdk/sdk/v0.0.70

15 **Auto-approval over ACP** NEW

65

Auto-approval settings are now honored over ACP, with a new `auto_approve` session config option (`AUTO_APPROVE_CONFIG_ID`) that ACP clients can set to approve all tool calls without prompting.

— Names exact config option and environment variable for ACP clients. cli-v3.0.50

16 **CLI usability improvements: history in TUI and accurate help output**

IMPROVED 65

`cline history` now opens inside the existing TUI with resume and delete actions instead of spawning a second view; `cline --help` now reports the real default values for `--config` and `--data-dir` paths.

— Names exact commands and flags whose output improved. cli-v3.0.48

17 **Cline Code desktop app launches for macOS** NEW 60

Releases Cline Code as a native macOS desktop app, signed and notarized for both Apple Silicon and Intel, for running and inspecting Cline agent sessions. The app checks for new versions on launch and every 2 hours, downloads them in the background, and prompts for a one-click restart; ignored updates apply on the next launch.

— Full update mechanism given; no config beyond installing the app. desktop-v0.0.2

18 **Onboarding and welcome flow improvements** NEW 60

Adds a Cline API key path to onboarding and enables cancellation of a pending browser sign-in. A first-run onboarding flow guides setup on launch, with a setting to replay onboarding and redesigned channel setup as expandable cards. The welcome screen shows a 'Connect a model' notice with one-click access to onboarding or model settings when no provider has credentials, reacting live and recognizing Bedrock/Vertex and keyless local endpoints as connected. Welcome suggestions adapt to what's in the opened folder instead of assuming a code project, non-git folders no longer show git jargon, and the folder picker offers manual path entry as a fallback.

— Several onboarding surfaces named but no single runnable command. desktop-v0.0.5 desktop-v0.0.4 desktop-v0.0.10 desktop-v0.0.11

19 **Session resilience across aborts and hub restarts** IMPROVED 60

Session context is now durable across aborts and hub restarts, so interrupted sessions resume with prior state rather than resetting. Queued prompts survive user-initiated and self-triggered aborts and are drained after a turn aborts itself; failed queued turns now emit `run.failed` instead of completing silently. Checkpoint diffs include files that were untracked at snapshot time, and checkpoints are picked up when git is initialized mid-session. Mid-stream network interruptions before any model output are retried automatically instead of failing the turn. `LiteLLM` requests now route through Chat Completions instead of the Responses API, enabling calls against LiteLLM proxies.

— Explains mechanism in detail but nothing for reader to trigger. desktop-v0.0.11 cli-v3.0.52 sdk/sdk/v0.0.72

20 **Plugin settings centralization and telemetry** IMPROVED

60

Plugin settings and contributions are now managed centrally in the hub with host-aware snapshots and atomic plugin toggles; empty install directories are no longer listed as installed, a source host no longer runs a foreign compiled plugin-sandbox bootstrap, and installed plugins display their real package names instead of all appearing as 'index'. Plugins can now emit telemetry through `ctx.telemetry` from both the subprocess sandbox and in-process execution.

— Names the `ctx.telemetry` API plugin authors can call directly. desktop-v0.0.11 v4.1.7 sdk/sdk/v0.0.72 desktop-v0.0.9 cli-v3.0.48 sdk/sdk/v0.0.67

21 **Hub upgrade coordination between installs** IMPROVED 60

Managed Hub daemons now upgrade directionally: when another Cline install ships a newer Hub build, the CLI attaches to the newer daemon and prompts you to update and restart (via `onHubUpdateRestart`) instead of installs repeatedly retiring each other's daemons. Upgrading the CLI automatically retires the running Hub daemon and respawns it on the new code. Yolo and sandbox sessions, which never attach to the shared Hub, are exempt from these interruption prompts.

— Names the callback but the process itself runs automatically. cli-v3.0.54

22 **Free-tier `cline-free` model support** NEW 60

Adds end-to-end support for `cline-free` models, labeled '(free)' in model pickers and priced at zero; hitting the free-tier usage limit shows a dedicated error card with the reset time instead of a generic failure. A new system tray icon also shows the count of running agent sessions alongside app status.

— Names the `cline-free` tier and its exact picker label. desktop-v0.0.7 v4.0.12 cli-v3.0.47 sdk/sdk/v0.0.66

23 **Ollama reliability and native AI SDK provider** IMPROVED 60

Ollama's response-start timeout now defaults to 5 minutes (up from 30 seconds), letting large models finish cold-loading before timing out; `requestTimeoutMs` can still override this explicitly. The model layer was upgraded to AI SDK 7 and Ollama switched to the native AI SDK provider, removing image-bearing request deprecation warnings.

— Names `requestTimeoutMs`, a usable override for slow models. desktop-v0.0.9 sdk/sdk/v0.0.70 desktop-v0.0.8 sdk/sdk/v0.0.69

24 **Claude Opus 5 and Moonshot Kimi K3 support** NEW 60

Adds Claude Opus 5 across the Anthropic, Claude Code, Bedrock, Vertex, Cline, and OpenRouter providers, including 1M context window variants; also adds Moonshot Kimi K3 model support.

— Names six providers and the 1M-context variant explicitly. v4.0.11

25 **CLI Hub auto-update deferral** IMPROVED 60

Auto-updates now defer installation until no CLI is attached to the Hub, with `cline update` still installing immediately and noting the update applies on next start.

— Names `cline update` as the immediate-install path. cli-v3.0.55

26 **Session history filter, favorites, pagination, and tray status** NEW 60

Session history can be filtered by origin — Desktop, CLI, extension, or scheduled — via a new filter control in the sidebar; sessions can be favorited and are ordered by most recent activity with consistent status-dot colors; history is paginated in ten-session pages, loading older history only when the end is reached; a new system tray icon shows app status and the count of running agent sessions; subagent and teammate runs from a session now appear with their status and results.

— Names the sidebar filter control and exact page size. desktop-v0.0.9 desktop-v0.0.7

27 **Composer and session UI quality-of-life improvements** IMPROVED 60

Images can be pasted directly from the clipboard into the composer. New keyboard shortcuts: `Cmd/Ctrl+N` for a new session and `Cmd/Ctrl+,` for settings; `Esc` stops the current turn. The composer displays a token usage ring for the active model's context window, including cumulative cost and a color change as the limit approaches. Settings, Sessions, Onboarding, and Diff views now load lazily for faster startup.

— Gives exact keyboard shortcuts usable immediately. desktop-v0.0.11 desktop-v0.0.10 desktop-v0.0.9

28 **Transcript UI overhaul with per-tool-call rows and diffs** IMPROVED 60

Every tool call now gets its own row in the transcript with its own icon, status, and expandable detail — commands render like a terminal (`$ bun test`) with captured output on expand, and edits show diffs inline per hunk. Running tool rows are highlighted in brand violet and settle to gray on completion; errors stay red. File diffs in chat rows and the diff panel now render through a shared syntax-highlighted renderer that follows the app theme. The thinking indicator remains visible during quiet stretches of a turn, such as while tool arguments are streaming.

— Detailed visual behavior described; nothing to configure. desktop-v0.0.12

29 **Generated media support: images, audio, and video** NEW 60

Supports inline image generation in the transcript for models with image-generation capability (model-driven image generation). The CLI forwards generated images as ACP agent message chunks with `type: image`, `data`, and `mimeType` fields; non-image generated media (audio, video) is represented as a `type: text` placeholder chunk. Adds a media player in the desktop app capable of playing generated audio artifacts and displaying generated video.

— Names exact ACP chunk fields for image and media output. desktop-v0.0.14 desktop-beta desktop-v0.0.14-beta.1

30 **`mode` field on `StartSessionInput`** NEW 60

Adds a `mode` field to `StartSessionInput` (values: `user`, `automation`, `subagent`, `team`) alongside the existing `source` field so sessions now record how they were initiated.

— Names the exact field and its four enum values. sdk/sdk/v0.0.70

THINNER COVERAGE BELOW

31 **New model providers: Chutes, Crusoe, Infomaniak, SCX.ai** NEW 55

Adds Chutes as a recognized provider alongside existing options like Anthropic, OpenAI, and OpenRouter; adds Crusoe to the refreshed model catalog with updated model lists and per-provider default models; adds Infomaniak and SCX.ai to the bundled provider and model catalog.

— Names four new providers; selecting one is the only action. desktop-v0.0.13 v4.1.10 cli-v3.0.55 sdk/sdk/v0.0.75 v4.1.4 desktop-v0.0.9 cli-v3.0.50

32 **Redesigned agent question prompts and run summaries** NEW 55

Redesigned agent question card supports explicit option selection, multiple-choice questions, and arrow-key and A–Z keyboard shortcuts, with animated reasoning and tool disclosures. Finished agent runs now collapse into a single expandable work summary (e.g. 'Worked for 4m 12s and made 14 tool calls'), keeping the final answer visible.

— Names the keyboard shortcuts for the new question card. desktop-v0.0.14 desktop-beta desktop-v0.0.14-beta.1

33 **Workspace-free chat, file attachments, and one-time routines** NEW 55

Supports workspace-free chat sessions — the desktop app can now start agent conversations without an open project folder. Adds one-time (non-recurring) routine scheduling alongside existing cron-based schedules, with in-app navigation to jump directly to a routine's run. Enables drag-and-drop file attachment directly onto the chat

input, displays image attachments inline in the chat transcript, and adds a custom overlay title bar with in-app navigation controls.

— Several capabilities named but no exact command surfaced. desktop-v0.0.4

34 **`onRetryAttempt` callback removed** BREAKING 50

Removes the never-invoked `onRetryAttempt` callback from `ApiHandlerOptions` and provider config; any code referencing it will break.

— Names removed callback and type; clear migration action. sdk/sdk/v0.0.66

35 **Unified reasoning controls across providers** IMPROVED 50

Reasoning settings (effort levels, thinking budget, enable/disable toggles) are mapped onto a shared resolution path and driven by the models.dev catalog across all AI SDK providers including Ollama, so behaviour is consistent without per-provider overrides; each model exposes exactly the reasoning options it supports, out-of-range budgets are clamped, and Anthropic's mandatory/impossible thinking modes are handled explicitly. Requests to disable reasoning are now respected everywhere.

— Explains mechanism broadly; no named flag to adjust directly. cli-v3.0.51 v4.1.5 desktop-v0.0.8 sdk/sdk/v0.0.67

36 **Automatic recovery from context overflow and empty responses** IMPROVED 50

Context-window overflow errors are now detected and auto-recovered — the runtime force-compacts with a deterministic strategy requiring no extra LLM call and retries once, surfacing an actionable message on terminal cases (nothing left to compact, or a retry that still overflows) instead of failing with a raw provider error. Empty model responses are now retried automatically at the model boundary on every provider (OpenRouter, Cline, OpenAI-compatible endpoints, and Ollama), though tool-call-only turns and turns that error or hit the token limit are never retried.

— Describes retry mechanism; entirely automatic, nothing to invoke. desktop-v0.0.9 cli-v3.0.50 sdk/sdk/v0.0.70 sdk/sdk/v0.0.69

37 **Yolo Mode toggle removed for auto-approve menu** BREAKING 50

Makes the auto-approve menu the single source of truth for unattended runs, removing the non-functional 'Yolo Mode' toggle; existing Yolo Mode or auto-approve-all setups are automatically migrated to auto-approving every action.

— Clear breaking change with described automatic migration path. v4.1.8

38 **ClinePass ACP provider** NEW 50

Adds `ClinePass` as a selectable ACP provider (`id: 'cline-pass'` , name: 'Sign in with ClinePass') alongside the existing Cline and ChatGPT Subscription options.

— Exact provider id and label given; directly selectable. `sdk/sdk/v0.0.68`

39 **'Enable R1 messages format' option removed** BREAKING 50

The 'Enable R1 messages format' option has been removed from the OpenAI-Compatible provider settings.

— Names the exact removed setting for migration awareness. `desktop-v0.0.8`

40 **Native macOS notifications** NEW 50

Adds native macOS notifications when a task finishes or needs input — configurable under Settings → Notifications.

— Exact settings path given to configure notifications. `desktop-v0.0.14`

41 **Cline Code Beta parallel install channel** NEW 50

Introduces a separate 'Cline Code Beta' app that installs alongside the stable build, updates from its own beta channel leaving stable installs unaffected, and self-identifies as beta in the sidebar, Settings, window title, and tray.

— Explains install and update behavior for the separate app. `desktop-v0.0.14` `desktop-beta` `desktop-v0.0.14-beta.1`

42 **Scheduled run reports gain execution context** IMPROVED 45

Scheduled run reports now include execution context — readable headers, schedule metadata, durations, and lifecycle error details — replacing sparse output; schedules also reuse saved provider settings instead of requiring separate configuration.

— Describes richer reports but no exact command or field shown. `desktop-v0.0.11` `v4.1.7` `cli-v3.0.52` `sdk/sdk/v0.0.72`

43 **Telemetry and diagnostics improvements** IMPROVED 45

Telemetry events now carry `device_id` and correct host identity (`host_plugin_version` , platform); `sdk.error` events are attributed to the model actually in use with undefined values stripped from event properties; adds telemetry for task lifecycle events, auth event metadata, and request IDs on SDK-pipeline events; the host plugin version is included in all telemetry events; application errors are now included in diagnostics, and telemetry configuration is baked into the sidecar at build time.

— Names specific telemetry fields; nothing for the reader to set. `sdk/sdk/v0.0.71`

desktop-v0.0.9 sdk/sdk/v0.0.67 sdk/sdk/v0.0.66 v4.0.11

44 **`meta/muse-spark-1.2-contributor` model added** NEW 45

Adds `meta/muse-spark-1.2-contributor` to the selectable model list on the Cline provider, alongside a refreshed model catalog.

— Exact model ID given; a straightforward catalog addition. v4.1.6 cli-v3.0.51 sdk/sdk/v0.0.71

45 **Dedicated Cline gateway provider** IMPROVED 45

Adds a dedicated Cline provider for the Cline gateway, replacing the generic OpenAI-compatible path, so extended thinking budgets and other gateway options now reach the wire for both `cline` and `cline-pass`.

— Names gateway ids but the change is internal wiring. sdk/sdk/v0.0.75

46 **Extension variant display in About page** NEW 45

Displays which extension variant is active — 'Legacy' or 'Next' — next to the version number in the settings About page for both bundles of the combined A/B rollout package.

— Names exact UI location to check the active variant. v4.1.2

47 **Voice input in composer** NEW 45

Adds voice input via a microphone button in the composer: speech is transcribed in real time using the configured provider and model, plus realtime voice sessions with UI refinements for the desktop app.

— Names the microphone button as the entry point. desktop-v0.0.14 desktop-beta desktop-v0.0.14-beta.1

48 **Queued-message manager and persistent update indicator** NEW 45

Adds a collapsible queued-message list above the composer showing a count, with per-turn options to edit, send immediately, or delete before submission. A new persistent sidebar update indicator shows the downloaded version with a one-click restart button that remains visible after dismissing the toast, and surfaces restart failures instead of silently ignoring them; the auto-update setting label is clarified to 'Keep CLI up to date', explaining it governs the `cline` terminal command rather than the desktop app.

— Describes two UI additions without a direct entry point. desktop-v0.0.6

49 **Skills in slash command menu** NEW 40

Shows skills alongside workflows in the slash command menu, and disambiguates commands that share a name instead of silently shadowing one with another.

— Names the UI location; thin on further mechanism. v4.1.4 desktop-v0.0.9

50 **Streaming markdown rendering improvements** IMPROVED 40

Streaming assistant markdown no longer flashes back to raw text — settled headings, links, and code stay rendered as new chunks arrive. Unifies the markdown rendering pipeline across the desktop app and cloud dashboard with chat-scaled headings, quieter code blocks with a hover copy button, and table cards.

— Describes rendering fix; nothing actionable for the reader. cli-v3.0.55 desktop-v0.0.14

51 **Clearer error surfacing across turns and providers** IMPROVED 40

Upstream provider errors forwarded through the gateway (including the Vercel AI Gateway) now surface the real message (e.g. "This model's maximum context length is 40960 tokens...") instead of a raw validation dump, Zod issue dump, or `[object Object]`. Failed turns now surface an error in the transcript with the underlying cause and a pointer to Settings → Models instead of failing silently.

— Describes better error text; nothing for reader to configure. desktop-v0.0.10 desktop-v0.0.8 sdk/sdk/v0.0.68

52 **Model catalog and request reliability fixes** IMPROVED 40

Custom model info for OpenAI-Compatible providers now carries over into the seeded model catalog, and unknown or removed legacy model IDs fall back to the default Cline model instead of failing. Requests to models without image support now substitute image content instead of failing, and MiniMax now inherits its default model from models.dev. Root-session persistence is now lazy — starting a runtime allocates the session id in memory without writing a database row, so closing before any user turn no longer leaves an empty history entry.

— Several small fixes named, but none are user-facing actions. sdk/sdk/v0.0.70 desktop-v0.0.8

53 **OpenRouter defaults to Claude Sonnet 5** IMPROVED 40

OpenRouter now defaults to Anthropic Claude Sonnet 5 (`anthropic/claude-sonnet-5`) as its model.

— Exact default model id named; nothing further to configure. desktop-v0.0.8 cli-v3.0.48 sdk/sdk/v0.0.67

54 **Agentic compaction becomes default strategy** IMPROVED 35

Agentic compaction is now the default context-compaction strategy across CLI, SDK, and desktop, replacing basic compaction as the fallback (previously required explicit opt-in), keeping long conversations within context limits more intelligently.

– States the default change; no opt-out flag is named. cli-v3.0.47 sdk/sdk/v0.0.66 desktop-v0.0.4

---

55 **Provider catalog and integration improvements** IMPROVED

35

The built-in provider list is now generated from models.dev, broadening out-of-the-box provider coverage; SAP AI Core integration now sets the metering header and uses the fetch adapter.

– Names two integrations but gives no configuration steps. sdk/sdk/v0.0.66

---

56 **Model-initiated plan-to-act switching removed** BREAKING

35

Removes model-initiated plan-to-act switching so that exiting Plan mode is always user-driven, never triggered mid-turn by the model.

– Clear before/after but no config surface named. v4.1.4

---

57 **Chat UI polish: tool icons, thinking time, timestamps, copy button**

IMPROVED

35

Tool-specific icons on tool disclosures, elapsed thinking time display, restyled reasoning sections, aligned timestamps, and a scroll-anchored conversation viewport; adds a copy button back to turn-final response rows.

– Lists cosmetic changes without deeper mechanism. desktop-v0.0.7 v4.1.7

---

58 **Vertex ADC token refresh via configured fetch** IMPROVED

35

Vertex AI credential/ADC token refreshes now use the configured fetch function/implementation, enabling authentication behind proxies and custom networking/transports.

– Explains the fix but nothing for the reader to configure. cli-v3.0.52 sdk/sdk/v0.0.72 desktop-v0.0.10

59 **Universal macOS binary** IMPROVED 35

Ships as a single universal macOS binary that runs natively on both Apple Silicon and Intel; existing per-architecture installs migrate automatically on next update.

— Migration is automatic; nothing for the reader to do. `desktop-v0.0.9`

60 **Font size slider in Settings** NEW 35

Adds a font size slider in Settings that scales the interface and applies before the window paints, eliminating the flash-to-old-size on launch.

— Simple UI addition with a clear settings location. `desktop-v0.0.13`

61 **Avatar/pet overlay** NEW 30

Adds a floating avatar/pet overlay (preview) to the desktop UI — a companion that reacts to active session state.

— Cosmetic preview feature with minimal operational detail. `desktop-beta` `desktop-v0.0.14-beta.1`

62 **Git branch tracking in TUI prompt** IMPROVED 30

Git branch shown below the TUI prompt now tracks external branch switches (from another terminal or editor) in real time.

— Small fix described; no action needed from the reader. `cli-v3.0.50`

63 **SSH remote environments (proof-of-concept)** NEW 25

Adds SSH remote environments as a proof-of-concept for connecting to and running sessions on remote development targets.

— Named as proof-of-concept only; no usage details given. `desktop-beta` `desktop-v0.0.14-beta.1`

┗—▷ BREAKING ON UPGRADE

! The 'Yolo Mode' toggle is removed; setups that had it enabled are migrated to auto-approving every action via the auto-approve menu.

! Model-initiated plan-to-act switching is removed: the model can no longer switch out of Plan mode on its own mid-turn; users must toggle to Act mode manually.

! The 'Enable R1 messages format' option has been removed from the OpenAI-Compatible provider settings.

! getShellArgs() is deprecated and replaced by getShellInvocation(); callers relying on getShellArgs() should migrate to getShellInvocatio

! The `onRetryAttempt` callback has been removed from `ApiHandlerOptions` and provider config — any code referencing it will break.

WAS THIS USEFUL? 👍 👎

All Hands AI OpenHands

9 RELEASES · 2026-07-21 → 2026-08-19

API ↗

OpenHands: AI-Driven Development

> OpenHands shipped its first public API surface alongside a wave of Automations, Canvas, and Activity Log improvements: Git Sync workflows, a context window usage meter with manual compaction, per-run LLM cost tracking, MCP server toggles, and SaaS gating for Agent Canvas.

└──▷ **WHAT SHIPPED · 23 FEATURES** most completely described first ? what's the number?

THINNER COVERAGE BELOW

01 **Public API surface published** NEW 55

OpenHands now publishes an OpenAPI spec covering 43 endpoints across three areas: Api (41 endpoints, supporting create, read, update, delete), Alive (1 read endpoint), and Health (1 read endpoint).

— Counts and areas named but no explicit paths or methods given 0.53.0

02 **Context window usage meter and manual compaction** NEW 55

Adds a context window usage meter, usage drawer, and manual compaction so practitioners can monitor and control context consumption in real time.

— Names distinct UI elements and control mechanism v1.13.0

03 **Activity Log cost display and export** IMPROVED 55

Adds per-run LLM cost display in the Activity Log and exports, giving teams visibility into spend at the individual run level, and adds activity log export capability to the interface.

— Names the Activity Log and export surface with concrete benefit v1.11.0 v1.10.0

04 **LLM profile pre-flight validation** NEW 45

Adds LLM pre-flight validation when saving LLM profiles, blocking misconfigured settings before they take effect.

— Explains behaviour and trigger point clearly v1.14.0

05 **Issue readiness gate for agent pickup** NEW 45

Adds a ready-for-dev issue readiness gate with type-specific criteria to validate issues before the agent picks them up.

— Explains purpose and criteria concept, no exact rules listed v1.13.0

06 **Typed child-conversation agent action** NEW 45

Adds a typed agent action for launching local or Cloud child conversations, enabling agents to programmatically spawn sub-conversations.

— Describes mechanism but no API name for the action v1.11.0

07 **Automations pane UX and manifest-driven structure** IMPROVED 45

Adds automation tag filter and recognition letting users filter the Automations pane by tag; keeps the Automations pane visible when empty so the panel is always accessible; ships a featured automations landing dashboard for discovering and launching automations; adds manifest-driven sub-pages to the automation interface enabling structured automation workflows; and drives the automation UI from the interface manifest for manifest-controlled automation flows.

— Groups thin Automations UI increments, all named individually v1.11.0 v1.10.0 v1.9.0

08 **Compact backend chooser in settings** IMPROVED 45

Adds a compact Cloud vs Agent-server backend chooser UI to the backends settings, making it faster to switch between hosting modes.

— Names exact settings location and the two backend modes v1.9.0

09 **MCP server enable/disable toggle** NEW 45

Adds a toggle on each installed MCP server's card to enable or disable it without uninstalling.

— Exact UI control and location named v1.8.0

10 **Git Sync automation page** NEW 40

Adds a Git Sync page under Automations for managing repository synchronization workflows.

— Clear UI location but no detail on sync mechanics v1.14.0

11 **Canvas default model changes** IMPROVED 40

Sets the Canvas default model to GLM 5.2, then later sets Kimi K3 as the default Canvas model and tags it as free.

— Names both models but no rollout mechanism v1.14.0 v1.10.0

12 **Faceted filter rail on Skills page** NEW 40

Adds a faceted filter rail to the Skills page for narrowing skills by category.

– Names exact page and filter behaviour v1.10.0

13 **Always-visible LLM selector and profile switching relocation** IMPROVED 40

Makes the LLM selector always visible in the UI and moves agent-profile switching into the tools menu.

– Names exact UI relocation destination v1.7.0

14 **Secret overwrite from edit form** IMPROVED 40

Allows overwriting an existing secret value directly from the secret edit form.

– Names exact form and action available v1.7.0

15 **Live agent activity in chat view** NEW 35

Shows live agent activity streamed directly in the chat view so users can follow what the agent is doing in real time.

– Describes behaviour but not underlying data source v1.9.0

16 **Persistent agent memory toggle** NEW 35

Adds a persistent agent memory toggle so agents can retain context across sessions.

– States effect but not scope of retained memory v1.7.0

17 **Agent Canvas gated behind SaaS authentication** BREAKING 35

Agent Canvas is now protected behind SaaS authentication, restricting access to authenticated users only.

– States the restriction but no migration guidance cloud-1.47.0

18 **Inline markdown artifact previews** NEW 30

Adds inline markdown artifact previews directly in the chat interface.

– Names the surface but no scope beyond that v1.13.0

19 **Domain-neutral extension-manifest host** NEW 30

Introduces a domain-neutral extension-manifest host to support loading extensions independently of the deployment domain.

– Technical capability named but no usage path given v1.9.0

20 **Canvas structured error outcomes** IMPROVED 25

Canvas now surfaces structured error outcomes, giving clearer feedback when agent tasks fail.

— No specifics on error format or where shown v1.14.0

21 **Client-side conversation archive** NEW 25

Adds a client-side conversation archive to organize and declutter past agent sessions.

— States capability but no detail on how archiving works v1.13.0

22 **Conversation view tag metadata** IMPROVED 25

Adds tag chips, overflow handling, and hovercard labels to the conversations view for richer conversation metadata display.

— Lists UI elements but no interaction detail v1.11.0

23 **Settings UI polish for navigation and Canvas updates** IMPROVED 15

Reorders the Customize navigation section for improved discoverability and polishes the Agent Canvas version update UI in Settings.

— Minor cosmetic changes with no further detail v1.11.0

WAS THIS USEFUL? 👍 👎

Daytona

7 RELEASES · seen 2026-08-19

API ↗

Daytona is a cloud development environment platform that provides standardized, reproducible coding workspaces for teams and remote development.

> Daytona published its full API reference (196 endpoints across 19 areas) and shipped a run of SDK/CLI additions across several 0.20x releases: pre-signed file transfer URLs, typed SDK error codes including git-specific codes, snapshot operations by name, outbound proxy support for sandbox egress, warm pool management, spot GPUs, and the promotion of sandbox forking and snapshot creation to stable.

└──▷ WHAT SHIPPED · 13 FEATURES most completely described first ? what's the number?

01 **Typed SDK error codes** NEW 85

Adds structured error responses with typed error codes across the SDK, enabling programmatic error handling by error type rather than string matching, and introduces `GIT_TRANSPORT_FAILED` and `GIT_REMOTE_REJECTED` typed error codes across all SDKs for finer-grained git failure handling.

Catch specific git transport failures in automation to distinguish a network error from a rejected push.

```
$ catch (err) {
  if (err.code === 'GIT_TRANSPORT_FAILED') { /* retry */ }
  if (err.code === 'GIT_REMOTE_REJECTED') { /* alert */ }
}
```

— Names exact error codes plus a runnable catch example

Snapshot operations by name and outbound… | Pre-signed file URLs and typed SDK errors | snapshot-20260819

02 **Outbound proxy for sandbox egress** NEW 80

Adds `outboundProxyUrl` parameter to the sandbox create call in the SDK, letting sandboxes route egress through a specified proxy — useful in air-gapped or regulated environments.

Create a sandbox that routes all outbound traffic through a corporate proxy — useful in air-gapped or regulated environments.

```
$ sandbox.create({ ..., outboundProxyUrl:
'http://proxy.corp.example.com:3128' })
```

— Exact config field with a runnable creation example

Snapshot operations by name and outbound…

03 **Snapshot operations by name and source filtering** NEW

70

Adds `sourceSandboxId` filter to snapshot list in the SDK, enabling callers to list only snapshots derived from a specific sandbox, and supports snapshot delete and activate by ID or name across the SDK and CLI, not just by ID.

— Names exact filter field and new lookup mode, no example given

Snapshot operations by name and outbound…

04 **Sandbox list filtering by autoDestroyAt** NEW

70

Adds an `autoDestroyAt` range filter to the sandbox list SDK method, enabling lifecycle-aware queries for sandboxes expiring within a time window.

— Names the exact filter field and its purpose Pre-signed file URLs and typed SDK errors

05 **Full API reference published** NEW

65

Daytona now publishes a full API reference covering 196 endpoints across 19 areas: Organizations (49 endpoints), Sandbox (47), Admin (24), Runners (12), Snapshots (8), Users (8), Api Keys (6), Docker Registry (6), and 11 more areas — Preview, Secret, Volumes, Jobs, Warm Pools, Webhooks, Health, Audit, Config, Object Storage, and Regions.

— Lists all areas and counts but no endpoint paths given 1.0

06 **Pre-signed sandbox file transfer URLs** NEW

60

Adds pre-signed download and upload URL generation for sandbox files, allowing direct S3-compatible file transfers without proxying through the API.

— Explains mechanism but no exact method signature given

Pre-signed file URLs and typed SDK errors snapshot-20260819

THINNER COVERAGE BELOW

07 **TypeScript SDK client-side HTTP timeout** NEW

55

Adds `DaytonaConfig.requestTimeoutMs` to the TypeScript SDK for configurable client-side HTTP timeouts.

— Exact config key named, straightforward to set Org members command and client-side HTTP…

08 **OpenTelemetry endpoint override for sandboxes** NEW 50

Adds `otelEndpointOverride` to sandbox creation for overriding the OpenTelemetry endpoint per sandbox.

– Names exact field, no example of use `Warm pool management and spot GPUs`

09 **Sandbox forking and snapshot creation stabilized** IMPROVED 45

Promotes sandbox forking and snapshot creation from experimental to stable in the SDK API, deprecating the previous experimental aliases.

– Notes stabilization and deprecation but no migration specifics

`Stable sandbox fork and snapshot creation`

10 **Optional modifiedAt in Python toolbox client** IMPROVED 45

Makes `FileInfo.modifiedAt` optional in the Python toolbox API client for compatibility with older runners.

– Names exact field and reason but small in scope `snapshot-20260819`

11 **Warm pool management APIs** NEW 30

Adds warm pool management APIs across SDKs.

– No endpoint names or mechanism given `Warm pool management and spot GPUs`

12 **CLI command to list organization members** NEW 30

Adds a CLI command for listing organization members.

– No command syntax or output details given `Org members command and client-side HTTP…`

13 **Spot GPU support for sandboxes** NEW 25

Adds spot GPU support for sandboxes.

– Bare mention with no configuration details `Warm pool management and spot GPUs`

WAS THIS USEFUL? 👍 👎

Cognition Devin Desktop

8 RELEASES · seen 2026-08-19 API ↗

Devin Desktop is an AI-powered code editor that helps developers write, debug, and deploy software with autonomous coding assistance.

> Devin Desktop shipped a documented public API (23 endpoints across 7 areas), introduced a Devin plugin system and expanded subagent support with custom agent files, added Plan mode with persistent Markdown plans and composable multi-level permission rules, and rounded out sandboxing, worktree, ACU-usage, and conversation-sharing capabilities across eight releases.

└──▷ WHAT SHIPPED · 42 FEATURES most completely described first ? what's the number?

01 Sandbox and CLI permission controls NEW

88

Adds CLI permission scopes so teams can enforce terminal allow/deny lists, and a `sandbox.excluded` config (available at user and team settings levels) with `allow` / `ask` / `deny` modes to run specific commands outside the sandbox, where excluded commands also bypass the sandbox proxy environment. Skill permissions frontmatter now applies to auto-approvals, and enterprise login policies are now enforced in the CLI.

Allow a specific command to run outside the sandbox entirely, bypassing both the sandbox and its proxy environment — useful for tools that break under the sandbox proxy.

```yaml
In Settings, open User or Team settings and set:
sandbox.excluded:
  allow:
    - my-special-tool
```

— Names config key, modes, and settings scope with example v3.4.22 v3.2.16

02 Subagent system with custom agent files NEW

85

Subagents can now call MCP tools directly, be configured with a default model, and be defined as flat `agents/<name>.md` files (per-agent directories also accept `AGENTS.md`, `agent.md`, and `agents.md`); a 'Subagents (Preview)' toggle in Devin settings enables the feature, and the Customizations panel's full-text search now includes a dedicated Subagents section.

Define a custom subagent for a specific task by creating a flat Markdown file in the agents/ directory.

```
$ # Create a custom subagent definition
mkdir -p agents
cat > agents/security-reviewer.md << 'EOF'
# Security Reviewer
You are a security-focused subagent. Review code changes for
vulnerabilities and suggest mitigations.
EOF
```

— Names file formats and toggle; missing capability limits v3.7.16 v3.6.21 v3.3.18 v3.2.16

03 **Share Conversation action for Devin Local** NEW 79

New 'Share Conversation' action for Devin Local conversations uploads a sanitized transcript (system prompts and tool definitions dropped, secrets redacted, paths normalized) and copies a team-visible link; accessible from the actions menu below a completed turn.

Share a completed Devin Local conversation with your team — transcript is auto-sanitized before upload.

📍 Open the actions menu below a completed turn › Share Conversation

— Explains sanitization steps and exact menu location v3.7.16

04 **Plan mode with persistent plan files** NEW 75

Plan mode now stores a persistent Markdown plan file at `~/.devin/plans/plan-<session>.md`, researches with read-only commands, and asks for approval before implementing — matching Cascade's plan workflow. Megaplan now works in Devin Local, switching the session into Plan mode and prompting the agent to plan first.

— Names the plan file path but not its full schema v3.6.21

05 **Migration tooling for hooks, skills and memories** NEW 73

Adds a command to migrate Windsurf hooks into Devin hooks across every workspace folder, and a new `Devin: Open Cascade Migration Wizard` command that opens a guided migration flow chaining hooks → skills → memories, with per-item opt-in and dry-run previews.

Run the Cascade Migration Wizard to migrate hooks, skills, and memories in one guided flow with dry-run previews before committing.

📍 Open the Command Palette (Cmd/Ctrl+Shift+P) and run: Devin: Open Cascade Migration Wizard

— Gives the exact command name and a runnable flow v3.6.21 v3.5.17

06 **Devin plugin system** NEW 70

Introduces a Devin plugin system for extending Devin Local, initially in preview and opt-in for enterprises. Devin Local Customizations gained a Plugins section listing loaded plugins and those offered by your repository, organization, or account, plus a refresh action for plugins. Plugins installed from the Customizations panel are added to personal plugins by default (available in Devin Cloud and on other devices); 'Install locally' limits the install to the current device.

— Explains install scope and locations, not plugin API v3.7.16 v3.6.21 v3.2.16

07 **Public REST API across 7 areas** NEW 68

Devin Desktop now publishes a documented API with 23 endpoints across seven areas: Sessions (7 endpoints) for creating and managing Devin sessions, Playbooks (5 endpoints), Knowledge (4 endpoints), Secrets (3 endpoints) for managing secrets and credentials, Attachments (2 endpoints) for file uploads, Auditlogs (1 endpoint), and Enterprise (1 endpoint) for enterprise-specific features and reporting.

— Names endpoint counts per area but not individual paths 1.0.0

08 **MCP server integration improvements** IMPROVED 68

MCP registry cache is now warmed during startup so MCP servers are ready sooner. MCP servers reporting 'Needs auth' now show an Authenticate button in the Devin Local MCP list, marketplace card, and detail page, which clears stored OAuth credentials and reruns the browser authorization flow. Each MCP server's logs are now separated into their own `MCP: <server>` output channel for cleaner observability.

— Names the log channel format and auth flow v3.7.16 v3.6.21 v3.3.18

09 **Worktree session support** IMPROVED 68

New worktree-backed sessions open instantly and stay interactive while the worktree is created in the background. Worktree sessions now show a Merge button to bring their changes back into the workspace, with an existing-worktree picker in the agent location selector's Local submenu. Starting an agent in a new worktree that fails now surfaces the git error instead of silently falling back to the main workspace, and 'Create New Terminal in Editor Area' now works inside worktrees.

— Names UI elements with clear before/after behavior v3.7.16 v3.6.21 v3.5.17

10 **Composable permission rules across levels** NEW 67

Composable permission rules now span enterprise, mode, user, project, and subagent levels — an explicit `deny: always` wins unconditionally, 'always allow' options appear only when the grant would actually take effect, and denials identify which layer denied them.

— Explains precedence mechanism but not config location v3.7.16

---

11 **Agent Command Center status bar hidden by default** BREAKING

67

The status bar in the Agent Command Center is now hidden by default; add `"workbench.statusBar.visible": true` to your user `settings.json` to restore it.

Restore the status bar in the Agent Command Center after it is hidden by default in this release.

```json
// In your user settings (settings.json)
{
  "workbench.statusBar.visible": true
}
```

— Exact config key, file, and value to reverse it v3.5.17

---

12 **Commit attribution config key** NEW

67

Adds an `attribution` option to the Devin Local config file; set it to `false` to suppress Devin mentions in commit messages generated by the agent.

Suppress 'Devin' attribution lines in commit messages generated by the agent.

```yaml
attribution: false
```

— Exact config key, value, and effect given v3.3.18

---

13 **ACU usage visibility** IMPROVED

65

Devin ACU usage is now displayed in the client and added to the `/usage` command. 'View usage' now opens eligible Devin users' personal analytics page directly. Session size chips use the same enterprise ACU thresholds as the web app, with auto-continuation billing notices appearing inline in the transcript.

Check how much ACU your current session or account has consumed without leaving the chat.

```
$ /usage
```

– Names the `/usage` command; thresholds unspecified v3.7.16 v3.5.17 v3.4.22 v3.3.18

14 **Modal settings overlay restore option** IMPROVED 64

Adds `workbench.editor.useModal` : `all` config key to restore the modal overlay for Settings and Keyboard Shortcuts, which now open as regular editor tabs by default.

Restore the modal overlay for Settings and Keyboard Shortcuts if your team prefers the old behavior.

```json
"workbench.editor.useModal": "all"
```

– Exact config key and value given, easy to apply v3.7.16

15 **Command approval editing and shortcuts** NEW 63

Editable command approvals let you click a command in a permission card to edit it before approving, or use the wand action to describe a change in plain language and have a fast model rewrite it for review. Adds keyboard shortcuts for permission requests (always-allow and reject), with shortcut hints shown on buttons and dropdown entries.

– Describes edit flow but not exact shortcut keys v3.6.21

16 **ACP browser preview for remote sessions** NEW 63

Remote agent (ACP) sessions can now open a browser preview: the agent proxies your local dev server, and captured elements and console output land in the agent's message box as pending context.

– Explains the mechanism but not activation steps v3.6.21

17 **Chat input and mention UX** IMPROVED 62

Images can now be copied from chat via the context menu. @ `-mention` pills in Cascade now show type-specific icons. Adds Rules to the @ `-mention` menu so manual-trigger rules can be pulled directly into a conversation, and selecting text inside a message transcript can be added to the chat input with the 'Add to chat' button or `⌘L` / `Ctrl+L` .

– Names the exact keyboard shortcut for one action v3.6.21 v3.5.17 v3.4.22

18 **Customizations panel search and organization** IMPROVED 60

The Hooks tab in Devin Local Customizations now lists configured hooks with their source and trigger events, and Devin Local customizations and the sidebar skills count now span all open workspace folders. 'Open customizations' is now available from the new-tab menu in an agent space and from a Devin Local session's sidebar context menu. The Customizations panel gains full-text search across skills, subagents, rules, hooks, plugins, and MCP servers, plus a new Subagents section.

— Lists searched surfaces but not query syntax v3.7.16 v3.6.21 v3.5.17

THINNER COVERAGE BELOW

19 **Queued message editing** IMPROVED 58

Queued messages can now be edited before they send, and editing or reverting a prompt preserves file, directory, skill, and terminal @`-mentions`; queued messages can also be edited by clicking the edit button or pressing ⬆ in an empty input to pull the last queued message back.

— Names the exact key to recall a queued message v3.7.16 v3.6.21

20 **Agent sidebar and session organization** IMPROVED 55

Adds a 'New session in space' option to the session kebab menu. The agent sidebar gains right-click context menus, double-click to rename, per-workspace filters/sort/grouping, greyed-out locked (read-only) sessions, a new-session shortcut hint, and sticky grouped spaces.

— Lists UI affordances without exact menu paths v3.6.21 v3.4.22

21 **Inline network access policy controls** NEW 52

Enables configuring a session's network policy and granting or denying network access requests inline from the chat, without switching to the web app. 'Grant access' on a network access request is now disabled with an explanation when an admin owns the session's network policy.

— Explains the flow but not the policy config surface v3.7.16 v3.5.17

22 **Codemap support in editor tabs and remote workspaces** IMPROVED 52

Codemaps now open in their own editor tabs in agent window mode, and codemap @`-mentions` now work in Devin Local. Codemaps and MCP configuration files now open from the remote machine when connected over WSL, SSH, or a dev container.

— Names remote connection types, not codemap generation v3.7.16 v3.6.21

23 **.devinignore file support** NEW 50

Adds `.devinignore` file support alongside `.windsurfignore` and `.codeiumignore` for controlling which files the agent processes.

— Names the file but not its syntax or precedence v3.1.7

24 **Devin Local only models disabled in Cascade picker** IMPROVED 50

Models marked 'Devin Local only' (including every GPT-5.6 variant) now appear disabled in Cascade's model picker with a tooltip pointing to Devin Local.

— Names the model family but not the full list v3.6.21

25 **Agent/Editor mode switching UI** IMPROVED 48

Adds the Agent/Editor mode switch to the collapsed-sidebar titlebar, unifies the search icon to open agent search, and enables the `Cmd+.` mode-toggle shortcut from the empty editor welcome panel.

— Names the shortcut and UI location, limited mechanism v3.2.16 v3.1.7

26 **Session duplication and forking** IMPROVED 48

Adds a 'Duplicate session' action to the response footer to branch off a conversation; 'Duplicate session' and 'Send as fork' now open the fork in a new tab, keeping the original conversation intact.

— Clear UI action but limited mechanism v3.7.16 v3.6.21

27 **Session performance and reliability** IMPROVED 47

Devin Cloud sessions now auto-reconnect when the network returns, and orphaned Devin ACP agent processes are now detected and cleaned up on startup, including on Windows. Long Devin Cloud sessions render, scroll, and type faster and stay responsive while streaming.

— States outcomes without the underlying mechanism v3.5.17 v3.4.22

28 **Diff review UI** IMPROVED 47

Edits produced in autonomous mode now produce reviewable diffs. The +X −Y diff summary on a collapsed agent group is now clickable and opens a multi-diff editor scoped to that group's files.

— Describes UI behavior without configuration options v3.7.16 v3.4.22

29 **Restricted Mode disables ACP agents** BREAKING 45

Cascade, Devin Local, and all other ACP agents are now unavailable while a workspace is open in Restricted Mode, and hooks no longer load or run there.

— States the restriction, not how to exit it v3.6.21

30 **devin.* settings honored** IMPROVED 45

Honors `devin.*` settings for gitignore access, completion mode, and auto-continue.

— Names the setting namespace, not exact key names v3.5.17

31 **Mid-turn revert controls** NEW 42

Revert buttons appear as soon as a prompt is sent; reverting mid-turn cancels the in-progress agent turn first.

— Describes behavior but not UI location v3.7.16

32 **Devin CLI renamed to Devin Local** IMPROVED 41

Renames the 'Devin CLI' settings section to 'Devin Local', adds a notification when Devin Local agent sign-in fails, and updates the bundled Devin Local agent to `v2026.5.26-8`.

— Lists the changes with little behavioral detail v3.1.7

33 **Editor tab and session card indicators** NEW 40

Pull request editor tabs now show a state-specific icon (open, draft, merged, or closed), and session-create cards show a Devin mode badge (Fast, Ultra, Fusion).

— Lists exact badge states but no interaction detail v3.6.21

34 **Session resume uses original working directory** IMPROVED 35

Resuming an agent session now uses its original working directory, enabling Claude sessions started in another folder to be reloaded.

— States the fix with no reproduction steps v3.5.17

35 **Unified session notifications** NEW 35

Adds a unified notifications setting that posts native OS notifications when any agent session finishes or needs your input.

— Names the setting broadly, not its exact key v3.6.21

36 **Bounded large-file reads via ACP** IMPROVED 35

Large-file reads through ACP are now bounded and paginated instead of repeatedly re-reading overflow files.

— States the fix but no size limits given v3.7.16

37 **Default agent/mode behavior for new sessions** IMPROVED 32

New users now default to agent mode, and new tabs default to Devin Local when no preferred agent is set, falling back to Cascade if unavailable.

— States the default with no override mentioned v3.6.21 v3.4.22

38 **VS Code base updated to 1.126** IMPROVED 28

Updated the base IDE to VS Code 1.126.

— Version bump with no further changelog v3.6.21

39 **Settings and MCP marketplace layout fix** IMPROVED 25

Settings and MCP marketplace pages now scroll across the full panel width and keep content centered on wide panels.

— Cosmetic layout fix with no further detail v3.1.7

40 **Fast Context support in Devin Local sessions** NEW 25

Adds Fast Context support in Devin Local sessions.

— Bare mention with no mechanism explained v3.5.17

41 **Timeline navigator for Local sessions** NEW 23

Adds a subtle timeline navigator for Devin Local sessions.

— No detail on how it works or is accessed v3.5.17

42 **Injected context excluded from session titles** IMPROVED 23

Injected context is no longer included in auto-generated session titles.

— Minor fix with no further mechanism described v3.3.18

└──▷ ALSO FROM THESE RELEASES

└──▷ BREAKING ON UPGRADE

! The status bar in the Agent Command Center is now hidden by default; set `"workbench.statusBar.visible": true` in user settings to restore it.

! Cascade, Devin Local, and all ACP agents are unavailable and hooks do not load or run while a workspace is open in Restricted Mode.

WAS THIS USEFUL? 👍 👎

SST OpenCode

8 RELEASES · 2026-07-20 → 2026-08-19 API ↗

The open source coding agent.

> OpenCode published its first public HTTP API (188 endpoints across 32 areas) and shipped a JSON session-export feature, an opt-in V2 desktop sidecar daemon, simplified xAI device-code login, and a string of desktop UI and provider improvements.

└──▷ WHAT SHIPPED · 9 FEATURES most completely described first ? what's the number?

01 **Opt-in V2 desktop sidecar daemon** NEW 95

Adds `OPENCODE_SIDECAR_V2=1` environment variable to initialize the desktop app from the V2 background daemon instead of the V1 utility sidecar, with daemon state persisted across desktop restarts. Also adds collapsible model provider sections in the V2 settings page, with collapse state persisted per server and search temporarily expanding matching sections without altering saved state.

Switch the desktop app to the V2 background daemon to try the new sidecar architecture.

```
$ OPENCODE_SIDECAR_V2=1 opencode
```

— Runnable env-var command plus mechanism and persistence behaviour given `v1.18.9`

02 **Session transcript export in desktop UI** NEW 90

Adds `Export...` to the 3-dot session dropdown menu, an `Export session` button in the Context tab, and a `session.export` command palette action (`/export`) to export full session transcripts as JSON from the desktop UI.

— Exact menu items, button and command palette action named `v1.18.15`

03 **Public HTTP API with 188 endpoints across 32 areas** NEW 75

OpenCode now publishes an OpenAPI spec covering `Session` (27 endpoints), `Sessions` (17), `Pty` (15), `Experimental` (13), `Tui` (13), `Instance` (12), `Mcp` (8), `Integrations` (7), and 24 more areas: `Permissions` , `Workspace` , `File` , `Global` , `Opencode Httpapi` , `Project` , `Projectcopy` , `Provider` , `Session Questions` , `Sync` , `Config` , `Control` , `Filesystem` , `Question` , `Permission` , `Providers` , `Commands` , `Controlplane` , `Event` , `Events` , `Messages` , `Models` , `Reference` , `Skills` . Many of these are marked experimental HttpApi routes.

— Names every area and endpoint count but no individual paths given `1.0.0`

04 **Desktop navigation and window management improvements** IMPROVED

65

Adds a right-click context menu on projects in the desktop Home view to open the project menu; keeps the macOS desktop app running after the last window closes and reopens a window when activated from the Dock; always shows the new session button regardless of state; and adds a `mod+shift+o` shortcut to open the project selector on the new-session page (disabled when no projects are available).

— Names an exact shortcut; other items are described UI behaviour v1.18.8 v1.18.10 v1.18.16

THINNER COVERAGE BELOW

05 **Desktop UI polish: notifications, tabs, terminal theme, review panel, prompt input**

IMPROVED

55

Improves toast notifications with better stacking, dismissal, and mobile layout; refines titlebar tab hover, active, and overflow states; syncs the embedded terminal theme with the app theme; improves the review panel so open file tabs stay aligned with the current diff view and improves review panel resizing and sticky controls; rewrites the v2 prompt input for more reliable command, context, shell, attachment, and history interactions; and adds much broader locale coverage across the desktop app.

— Several concrete UI areas named but no exact controls or paths v1.18.4 v1.18.10 v1.18.15

06 **MCP client SDK upgraded to v2** IMPROVED

45

Upgrades the MCP client SDK to v2, improving compatibility with newer MCP servers and OAuth flows.

— States the upgrade and its benefit without further mechanism v1.18.8

07 **Device-code login flow for xAI** IMPROVED

45

Simplifies xAI authentication to a single device-code flow, eliminating OAuth confusion in headless and remote environments.

— Describes the change but gives no exact command or flag v1.18.14

08 **Lenient top-level config parsing** IMPROVED

45

Unknown top-level config fields are now silently ignored instead of causing config parsing failures.

— Explains the behaviour change but names no specific config keys v1.18.16

09 **Automatic Modal model discovery** NEW

35

Automatically discovers available Modal models, removing the need to manually configure them.

— Bare description of the discovery behaviour, no config detail `v1.18.10`

WAS THIS USEFUL? 👍 👎

Cotool

15 RELEASES · seen 2026-08-19 NOTES ↗

> Cotool shipped a full first-class Alerts system for security triage, Hunt V2 continuous threat-intelligence hunting, and Response Agents as Code with GitHub GitOps sync, alongside dozens of new integrations (StepSecurity, Zscaler, Darktrace, OpenCTI, HackerOne, Bugcrowd, Huntress, and more), new AI models including GPT-5.6 and Claude Opus 5 with Auto Model Routing, and a redesigned navigation.

└──▷ WHAT SHIPPED · 63 FEATURES most completely described first ? what's the number?

01 **Slack integration for agent workflows** IMPROVED 85

Slack agent workflows gained run-button configuration and support for user replies, sandbox file attachments in Slack messages, custom Slack app integration setup, configurable timeouts for unanswered Slack confirmations, a Slack reply scope setting so triggers respond to anyone in a thread or only Cotool users, and Slack emoji-reaction triggers so reacting to a message runs an agent.

— Names six distinct Slack behaviors without exact API config v0.63.0 v0.58.0 v0.56.0 v0.55.0 v0.54.0 v0.53.0

02 **Response Agents as Code** NEW 80

Response agents can be defined and synced via GitHub GitOps sync, including sample YAML agents and managed-agent protections to prevent accidental edits. Later releases added API-based schema validation, canonical YAML imports, support for multiple GitHub sync repositories per organization, and made webhook URLs and secrets visible on GitOps-managed triggers.

— Names GitOps sync, YAML format, validation, and multi-repo support v0.59.0 v0.57.0 v0.56.0

03 **Threat intelligence sources and management** IMPROVED 80

Threat intelligence ingestion adds cross-source deduplication, huntability filtering, and Socket Blog support, plus IOC list views and a MISP Hunt intel source. Sources are consolidated with MISP sync cleanup and retries for failed API sync results, and an OpenCTI integration enriches investigations with threat intel. Recorded Future, Silobreaker, and TAXII 2.1 were added as sources, Recorded Future was expanded with alert and alert-rule search plus full alert detail retrieval, and Silobreaker gained broader threat-intelligence research.

— Names many sources and specific dedup/search features, no config keys v0.64.0 v0.61.0 v0.60.0 v0.58.0 v0.53.0 v0.52.0

04 **First-class Alerts system for triage** NEW 80

First-class Alerts for security triage include an alert inbox, detail pages, activity timelines, comments, assignments, dispositions, bulk actions, and response-agent triage. The Alerts page later gained contextual `Cmd+K` status commands for bulk selection and faster indexed alert search.

– Names concrete alert-management surfaces and a keyboard shortcut v0.60.0 v0.59.0 v0.57.0

05 **Hunt V2 continuous threat intelligence** NEW 80

Hunt V2 continuously assesses incoming threat intelligence against your environment, hunts for exposure, and raises alerts with a verdict when a threat is relevant to your organization. The `/hunt-intel` slash command ingests a threat report URL through the Hunt pipeline, with a dedicated timeline in chat.

– Names the verdict mechanism and the exact slash command v0.60.0 v0.59.0

06 **Detection hit inspection and evidence workflow** IMPROVED 75

Detection agents can inspect hits and call tools directly from hit context, with subagent-based denoising to reduce noisy findings, step reordering, and a redesigned detection hits interface. Evidence capture now cites tool calls and shows richer query results, with evidence surfaced on hit detail pages, agent-type filtering, cleaner hit displays, and unified run pagination. Suggested detection agents are unified with detection-rule suggestions, adding investigation-plan review, direct acceptance into the detection builder, and bulk dismiss for suggestions.

– Covers mechanism and UI surfaces across six releases without exact endpoints v0.61.0 v0.59.0 v0.53.0 v0.52.0 v0.51.0 v0.50.0

07 **New AI chat models and defaults** BREAKING 75

Claude Opus 4.8 and GPT-5.5 were added, with GPT-5.5 becoming the default chat model — a breaking change for workflows relying on the prior default. GPT-5.6 models followed, with GPT-5.6 Sol as the new default, and Claude Opus 5 plus open-weight models Kimi K3 and GLM 5.2 were added with the same data residency and ZDR guarantees as proprietary models.

– Names every model and flags the default-change breaking impact v0.60.0 v0.59.0 v0.57.0

08 **Sumo Logic detection authoring and validation** IMPROVED 70

Sumo Logic query validation now uses a formal parser for improved detection query feedback and reliability, environment mapping supports editable descriptions, runtime

variable updates, and larger mappings, and compiler feedback adds stronger parsing advisories and empty-result validation.

– Names formal parser, editable mappings, and validation additions v0.52.0 v0.51.0

09 **Linear integration for detection and ticketing** NEW 70

Detection agents can output generated detections directly as Linear tickets, and a Linear agent trigger endpoint starts Linear-linked agent workflows programmatically. Later additions include a Linear team issue listing tool, Linear label support, full-text Linear issue search with team, status, archive, and comment filters, and parent-child issue updates.

– Names endpoint, filters and labels but no exact API path v0.64.0 v0.62.0 v0.59.0 v0.54.0 v0.52.0

10 **Sub-agent execution visibility and handoff** IMPROVED 70

Sub-agent UX gained progress previews, drawer breadcrumbs, and clearer nested agent timelines, a sub-agent timeline drilldown in the execution panel, tool chips that link to the agent details page, reusable execution receipts that preserve successful tool calls and output files to reduce duplicate lookups, and live progress from delegated agents shown in chat and agent timelines.

– Names five distinct sub-agent UX and handoff mechanisms v0.64.0 v0.61.0 v0.59.0 v0.56.0 v0.53.0

11 **Darktrace integration and breach actions** NEW 70

A Darktrace integration supports network detection and response investigations, later expanded with actions to acknowledge, reopen, and comment on model breaches, and webhook triggers so response agents run immediately on model-breach and AI Analyst alerts.

– Names specific breach actions and webhook triggers v0.64.0 v0.62.0 v0.55.0

12 **Huntress, Railway, Kolide, Cloudsmith integrations** NEW 70

New Huntress, Railway, Kolide, and Cloudsmith integrations were added. Huntress was later expanded for MSP and MSSP accounts with optional organization scoping, Kolide gained expanded report queries and improved multi-value parameter handling, and Cloudsmith setup added connect-time credential validation and owner-aware tool errors.

– Names four integrations and three concrete extensions v0.60.0 v0.61.0 v0.62.0

13 **StepSecurity integration for GitHub Actions** NEW 70

A StepSecurity integration supports investigating GitHub Actions security posture, runtime detections, network and process activity, policy evaluations, compromised components, and supply-chain threats.

— Names six distinct investigation surfaces v0.61.0

14 **Zscaler integration for ZIA, ZPA, ZDX** NEW 70

A Zscaler integration covers ZIA, ZPA, ZDX, and Client Connector tools, with detection-rule sync, alert triggers, and permission validation.

— Names four Zscaler products and three integration mechanisms v0.64.0

15 **GitHub tooling for agents** IMPROVED 65

Agents gained GitHub commit history and blame tools for repository investigations, the ability to create draft GitHub pull requests, support for uploading sandbox-generated files via GitHub tools, and multi-organization GitHub App installation scoping.

— Names four distinct GitHub capabilities without exact commands v0.63.0 v0.62.0 v0.60.0 v0.53.0

16 **New alert statuses: Dismissed, Duplicate, Expired** NEW 65

A Dismissed status (human-only) archives alerts without closing them, a Duplicate alert status was added, and an Expired status with 14-day inactivity checks surfaces and manages stale alerts.

— Names each status and Dismissed's exact behavior verbatim v0.61.0 v0.60.0 v0.58.0

17 **Model routing and settings redesign** NEW 65

Auto Model Routing lets Cotool automatically select the model backing a chat or agent task. Model settings were redesigned with a searchable provider catalog, clearer Cotool Auto routing, and dedicated custom endpoint management, and agents can now receive guided, user-approved model upgrades with supporting research for each recommendation.

— Describes routing and settings redesign without exact UI path v0.62.0 v0.61.0 v0.60.0

18 **Barracuda, OX Security, Supabase integrations** NEW 65

A new Barracuda integration supports investigating and remediating email threats, a new OX Security integration searches applications, artifacts, and SBOMs and investigates issues and attack paths, and a new Supabase integration covers organization and project security posture, configuration review, and project logs.

— Names three integrations and their concrete scopes v0.63.0

19 **Navigation and page layout redesign** IMPROVED 65

Navigation and page layouts were redesigned around Home, Detect, Hunt, Respond, Knowledge, Platform, and Settings sections, with consistent breadcrumbs and backward-compatible existing links.

— Names all seven new sections and compatibility guarantee v0.64.0

20 **Detection and response notification controls** NEW 60

Organization-level notification settings now control detection and response output delivery, and detection notification severity filters let teams control which detection hits trigger downstream destination notifications.

— Names settings and filters but no exact config location v0.57.0 v0.53.0

21 **Audit log filtering and coverage improvements** IMPROVED 60

Audit logs gained richer filters, user and tool filtering, native tool hiding, and infinite scroll, plus logging for agent status changes and a distinction between API agent executions and interactive runs.

— Names filter types and log distinctions, no query syntax v0.62.0 v0.60.0 v0.53.0

22 **Bugcrowd and HackerOne vulnerability intake** NEW 60

A Bugcrowd integration adds webhook triggers for vulnerability intake and response automation, and a HackerOne connector adds webhook-triggered alert intake for vulnerability response workflows.

— Names webhook trigger mechanism for both connectors v0.57.0 v0.56.0

THINNER COVERAGE BELOW

23 **ExtraHop, Semgrep, ClickHouse, Obsidian integrations** NEW 55

New ExtraHop RevealX, Semgrep, ClickHouse, and Obsidian Security integrations were added, with Obsidian later gaining support for closing alerts directly in the tool.

— Names four integrations plus one extension, no setup detail v0.58.0 v0.53.0

24 **Persistent Agent Filesystem** NEW 55

A Persistent Agent Filesystem lets agents save and reuse files across runs, enabling stateful workflows between executions.

— Explains mechanism but no file size limits or API given v0.54.0

25 **Jira Service Management output and auth** NEW 55

Jira Service Management alert creation is available as an agent output destination, and Jira now supports a service account authentication option.

— Names both Jira capabilities without setup steps `v0.58.0` `v0.54.0`

26 **Executive dashboard time-window enhancements** IMPROVED 55

The executive dashboard gained URL-encoded time windows, a one-year preset, and a refined date picker.

— Names concrete UI changes with no navigation path `v0.54.0`

27 **Canonical audit-log event catalog endpoint** NEW 55

A canonical audit-log event catalog endpoint, with documentation, gives practitioners a single authoritative source for audit event types.

— Names a documented endpoint but not its path or method `v0.59.0`

28 **Integrations directory redesign** IMPROVED 55

The integrations directory was redesigned with search, category and connection-status filters, and clearer integration details.

— Names filter types and a UI redesign, no exact path `v0.61.0`

29 **Durable agent runs across restarts** IMPROVED 55

Agent runs, delegated sub-agents, tool calls, waiting prompts, and handoffs are now durable across worker restarts.

— Names five durable components without failure-recovery detail `v0.62.0`

30 **Private skills, bulk import, and namespaces** NEW 50

Private skills scope organization-specific workflows to specific users and agents, with skills pages showing private skill visibility and invocation counts. Skill libraries can be bulk-imported from folders, and skills now live in separate personal and organization namespaces.

— Names scope and namespaces but no exact UI path or command `v0.59.0` `v0.52.0` `v0.50.0`

31 **Agent Skills as code** NEW 50

Skills can now be managed and synced from GitHub, mirroring the existing agents-as-code workflow, so skill definitions live in version control alongside agent definitions.

— Names the sync mechanism but no exact repo or config path `v0.58.0`

32 **Agent run reliability and reporting** IMPROVED 50

Agent runs triggered via API now have rate limiting and clearer reporting, and acceptance criteria evaluation now covers errored executions so failures still produce useful feedback.

– Describes behavior change but no configurable limits given `v0.51.0` `v0.50.0`

33 **FireHydrant and Glean integrations** NEW 50

A new FireHydrant integration brings incident context into agent workflows, and a new Glean integration provides search and document access as a knowledge source for agents.

– Names both integrations and their purpose, no setup steps `v0.51.0`

34 **Configurable SCIM provisioning** NEW 50

Configurable SCIM provisioning is now supported, with public SAML and SCIM identity provider documentation.

– Names SAML/SCIM support and docs, no exact config keys `v0.54.0`

35 **CrowdStrike tool expansions** IMPROVED 50

The CrowdStrike tool gained Adaptive Shield (Falcon Shield) SaaS security support and CrowdStrike Spotlight vulnerability-management actions.

– Names both added capabilities without action detail `v0.60.0` `v0.55.0`

36 **Slash commands in chat** NEW 50

Chat gains slash commands with persisted commands for detection agents and a refreshed command header in the chat input.

– Describes command persistence, no full command list `v0.59.0`

37 **Terminal-style rendering for CLI tool calls** NEW 50

CLI-based tool calls such as `gcloud` and `aws` now render with terminal-style visuals in chat.

– Names the exact tools rendered, no interaction detail `v0.60.0`

38 **Skill version history** NEW 45

Skills gain version history with the ability to inspect and restore earlier versions.

– Clear restore action but no exact UI path `v0.62.0`

39 **Exa web search agent tool** NEW 45

Exa web search is now a selectable agent tool for retrieving fresh web context during agent runs.

– Names the tool and purpose but no config detail `v0.50.0`

40 **Structured output generation and rendering** IMPROVED 45

Structured output generation now retries after interrupted generation for more consistent results, and rendering has been refined with schema previews and a cleaner detection output viewer.

— Names retry and viewer changes but no retry limits given v0.52.0 v0.51.0

41 **KnowBe4 PhishER integration** NEW 45

A KnowBe4 integration adds PhishER GraphQL support for phishing triage workflows.

— Names GraphQL support but no query examples v0.55.0

42 **Agent versioning to revert changes** NEW 45

Agent versioning tracks and lets teams revert agent changes over time.

— Clear revert action but no version limit or UI path v0.55.0

43 **Tines and VirusTotal investigation tooling** IMPROVED 45

New Tines record retrieval tools and expanded VirusTotal relationship pagination support deeper investigations.

— Names both tools without usage specifics v0.57.0

44 **Notifications page redesign and escalation filters** IMPROVED 45

The notifications page was reworked with tabs and added escalated-alert notifications, which later gained a minimum-severity filter.

— Names tabs and severity filter without exact settings path v0.63.0 v0.58.0

45 **Per-organization session length setting** NEW 45

A per-organization session length setting is now available in Settings.

— Names the setting location but no default or range v0.63.0

46 **Custom RSS feed URL validation** IMPROVED 40

Custom RSS feed setup now validates the URL before feeds are saved.

— Small but exact behavior described v0.52.0

47 **Inline chart rendering in chat** NEW 40

Charts now render inline in the chat timeline with light and dark theming.

— Concrete UI feature, no interaction detail v0.55.0

48 **MITRE coverage view** NEW 40

A MITRE coverage view supports environment-level threat model analysis.

— Names the view but no navigation or scope detail v0.57.0

49 **Output destination test-send from UI** NEW 40

Output destinations can now be tested directly from the UI by sending an example payload.

— Clear UI action, no navigation path given v0.58.0

50 **Response agents from chat or templates** NEW 40

Response agents can now be created directly from chat or from templates.

— Clear starting points but no template list v0.64.0

51 **Bulk deletion for response agents** NEW 35

Response agents can now be bulk-deleted to streamline cleanup at scale.

— Clear action but no scope or limit details v0.51.0

52 **Spacelift integration for IaC investigations** NEW 35

A Spacelift integration supports infrastructure-as-code investigations directly from agent workflows.

— Thin description of what it enables v0.54.0

53 **Job dependencies for scheduled jobs** NEW 35

Job dependencies let scheduled and chained jobs execute in the correct order.

— States behavior with no configuration syntax given v0.54.0

54 **Schema-free response output delivery** IMPROVED 35

Response output delivery can now work without a structured schema, making output destinations more flexible.

— Describes flexibility gained without concrete example v0.56.0

55 **Enhanced agent improvement generation** IMPROVED 35

Agent improvement generation now allows tool suggestions and richer review flows.

— Thin description of review flow changes v0.56.0

56 **Minor UI polish: tags and tooltips** IMPROVED 35

Agent tag counts are now clickable in the UI, and status-explainer tooltips were added to Threats group headers.

— Two small named UI tweaks, no deeper mechanism v0.60.0

57 **Instruction-aware integration recommendations** NEW 35

The agent builder now surfaces instruction-aware integration recommendations directly during agent setup.

— Thin description of recommendation mechanism v0.62.0

58 **Notion integration nested page content** IMPROVED 35

The Notion integration now retrieves complete nested page content.

— Thin single-clause extension with no further detail v0.64.0

59 **Wiz investigations enhancements** IMPROVED 30

Wiz investigations are enhanced by consolidating agent tools and streamlining principal activity handling.

— Thin description with no concrete mechanism v0.52.0

60 **Datadog detection authoring improvements** IMPROVED 30

Datadog detection authoring is improved with better rule generation, validation, and helper coverage.

— Generic improvement claim without specifics v0.52.0

61 **Provider-native conversation compaction** IMPROVED 30

Provider-native compaction preserves more useful context in long conversations.

— Names the mechanism but no size or trigger threshold v0.62.0

62 **Reply and follow-up with response agents** NEW 25

Users can now reply to and follow up with built-in response agents.

— Bare description of new capability v0.55.0

63 **ChartHop integration for org context** NEW 25

A ChartHop integration brings people and organization context into investigations.

— Bare integration description v0.56.0

└──▷ **BREAKING ON UPGRADE**

! GPT-5.5 replaces the previous model as the default chat model; existing workflows relying on the prior default will now use GPT-5.5.

WAS THIS USEFUL? 👍 👎

Amazon Kiro

10 RELEASES · seen 2026-08-19 NOTES ↗

> Kiro's biggest addition this window is Claude Opus 5 with a 1M-token context window and cross-region inference, alongside a new open Agent Plugin format for installable Powers and a Cloud Sessions preview letting CLI and IDE sessions run in a managed cloud sandbox. The CLI also gained `/tangent` side-conversations, `/voice` dictation, and spec-review tooling, while nested `AGENTS.md` support, Agent Focus Mode UI upgrades, and a string of smaller editor, permission, and terminal improvements rounded out the release.

└──▷ **WHAT SHIPPED · 22 FEATURES** most completely described first ? what's the number?

01 Cloud Sessions preview in CLI and IDE NEW

95

Kiro CLI adds a `--cloud` flag (`kiro-cli chat --cloud`) to run a session in a managed cloud sandbox instead of locally, a `--repo` flag (and `/repo` picker) to attach repositories to a cloud session, and a `--resume-id` flag to reconnect to a cloud session from any machine after disconnecting while the agent keeps working; cloud and local sessions appear together in the session picker UI. The Kiro IDE's Agent Focus Mode adds the same Cloud Sessions preview, letting sessions run in the cloud alongside local ones with selectable repositories kept visible below chat, using the same agent and model selectors as local sessions.

Start a cloud session, attach a repo, then disconnect and resume later from another machine.

```
$ kiro-cli chat --cloud --repo my-org/my-repo
# later, from any machine:
kiro-cli chat --resume-id <session-id>
```

— Exact flags and a runnable resume example are provided.

Cloud Sessions and Agent Focus Mode Upgr… Cloud Sessions Preview and Smarter Comma…

02 /tangent side-conversations in CLI NEW

95

Adds `/tangent <name>` to branch a named side-conversation that inherits the full conversation history, letting you explore freely without polluting the main thread; `/tangent ls` browses the full conversation tree in a visual picker; `/tangent root` jumps straight back to the main conversation from any depth. All `/tangent` features are available in V3 mode via `kiro-cli --v3` .

Branch into a named side-conversation mid-task to look something up, then return to the main thread without losing context.

```
$ kiro-cli --v3
# inside the session:
/tangent research-auth-flow
# ... explore freely ...
/tangent root
```

Visualise your full conversation tree to find and jump to any branch you created.

```
$ /tangent ls
```

— Exact subcommands and runnable examples are provided.

Tangent Side-Conversations and Per-Tool …

03 **Claude Opus 5 model added** NEW 80

Claude Opus 5 is now available in the Kiro IDE, CLI, and Web model selector for Pro, Pro+, Pro Max, and Power customers in `us-east-1` and `eu-central-1` with cross-region inference. It supports a 1M token context window with a 2.2x credit multiplier, brings improved multi-agent coordination with fewer conflicts between parallel agents, adds self-verification and iterative refinement that reduces stubs and placeholders in generated code, and improves code review with higher real-bug detection and lower false positives even at reduced effort settings.

— Concrete numbers and regions given; no exact selection steps.

Claude Opus 5 Now Available

04 **Spec phase review with Ctrl+X** NEW 80

Adds `Ctrl+X` at spec phase checkpoints to open a review screen where you can read phase documents and stage line comments, sending all staged comments as one revision request when the checkpoint is answered.

Review a spec phase document and request targeted edits without leaving the checkpoint screen.

> 📍 Ctrl+X at a spec phase checkpoint to open the review screen, add line comments on sections to change, then answer the checkpoint question to send all staged comments as one revision request.

— Exact shortcut and full workflow steps are given.

Voice Mode, Opt-In Cloud Sessions, and S…

---

05 **Cloud Sessions admin setting renamed and made opt-in** BREAKING 75

The IAM Identity Center / Kiro console admin setting for cloud sessions was renamed from 'Kiro Web (Preview)' to 'Cloud Sessions (Preview)' — organizations that had the prior setting remain enabled, but new organizations must explicitly opt in. It was later changed to default to disabled: administrators must now explicitly enable 'Cloud Sessions' in the Kiro console, and organizations that never configured the toggle will find cloud sessions unavailable until an admin opts in.

— Names the exact setting and its before/after states.

Voice Mode, Opt-In Cloud Sessions, and S… Cloud Sessions Preview and Smarter Comma…

---

06 **Nested AGENTS.md support** IMPROVED 75

Kiro supports `AGENTS.md` files placed at any level of a workspace directory tree, allowing agent instructions scoped to each subdirectory. This was extended so `AGENTS.md` steering files can live anywhere in the workspace tree — not just the workspace root or `~/.kiro/steering/` — letting steering context sit next to the code it describes.

— Names the exact default paths and the extended scope.

Voice Mode, Opt-In Cloud Sessions, and S… Nested AGENTS.md and Long-Session Effici…

---

07 **Voice dictation via /voice** NEW 75

Adds a `/voice` command (also triggered by holding `Space`) to dictate prompts with on-device Whisper transcription — no audio leaves the machine and no cloud API key is required.

Dictate a prompt hands-free when typing is inconvenient — transcription runs locally so no audio is sent to the cloud.

— Exact command and privacy mechanism named, with an example.

Voice Mode, Opt-In Cloud Sessions, and S…

---

08 **Agent Focus Mode session and UI improvements** NEW 70

Agent Focus Mode gains a Pin Session action to keep a session pinned at the top of its section in the session rail, an Open with Kiro CLI action to resume an existing session directly in Kiro CLI, an update row in the rail footer with a Check for Updates button in settings, bulk session migration for existing sessions, and spec documents/task rows that now open directly in the editor. It also adds a collapsible icon-only view for the sessions rail, in-place attention cards that surface agent questions and action buttons next to the chat without a context switch, and back/forward navigation across sessions and settings from the titlebar via `Cmd/Ctrl+[` and `Cmd/Ctrl+]`.

— Lists concrete UI actions and shortcuts but each is thinly described.

Cloud Sessions and Agent Focus Mode Upgr… Agent Plugin Support and Session Pinning

---

09 **Guided description step in /spec new** IMPROVED 65

Adds a guided description prompt to `/spec new`: the agent now asks what the spec should cover before drafting requirements, using the description as ground truth instead of inferring from the spec name alone.

— Names the exact command and describes the behavior change.

Spec Drafting and Plan Auto-Execution

---

10 **Agent Plugin format support for Powers** NEW 60

Adds support for the open Agent Plugin format, letting users install 'Powers' — plugins that bundle skills and MCP servers — from a local folder or a GitHub URL, shareable across compatible agent tools.

— Names the format and install sources, but no exact command shown.

Agent Plugin Support and Session Pinning

THINNER COVERAGE BELOW

---

11 **Editor context menu and Ask Kiro to Fix** NEW 50

Adds a 'Kiro' submenu to the editor context menu for quick access to Kiro actions from within the editor, and an 'Ask Kiro to Fix' quick-fix action on errors and warnings that surfaces AI-assisted remediation inline.

— Names UI entry points but no further mechanism given.

Editor Actions, Guided Hook Creation, an…

---

12 **Per-tool token breakdown in /context** IMPROVED 50

Extends `/context` with a per-tool token breakdown so you can see exactly what is consuming your context window.

– Names the command but shows no example output. Tangent Side-Conversations and Per-Tool …

13 **Agent terminal shell handling on Windows** IMPROVED 45

Kiro's agent terminal on Windows was first pinned to PowerShell, then changed so agent commands run in the user's configured Windows terminal profile instead of a default shell.

– States the change but no config key or flag is named.

Permission Improvements and Configured W… Cloud Sessions and Agent Focus Mode Upgr…

14 **Plan mode auto-executes approved plans** IMPROVED 45

Plan mode now auto-executes the approved plan immediately after approval, removing the manual mode-switch step.

– Clear behavior change but no flag or command named.

Spec Drafting and Plan Auto-Execution

15 **Guided hook creation form** IMPROVED 35

Adds a guided form for creating hooks, replacing the need to author hook configuration manually.

– Describes the change but not the form's fields or location.

Editor Actions, Guided Hook Creation, an…

16 **Slash command substring matching** IMPROVED 35

The slash command menu gains substring matching, making commands reachable without typing a prefix.

– Explains the behavior but gives no example. Cloud Sessions Preview and Smarter Comma…

17 **Code OSS upgraded to v1.108.2** IMPROVED 30

Upgrades the underlying editor to Code OSS v1.108.2.

– Just a version bump, no accompanying feature detail.

Editor Actions, Guided Hook Creation, an…

18 **Long-session performance improvements** IMPROVED 25

Kiro preserves more context during long sessions, reducing loss of conversation state over extended interactions, and lowers idle resource consumption during inactive periods.

– Describes outcomes only, no metrics or mechanism.

Nested AGENTS.md and Long-Session Effici…

19 **Replies in IDE display language** IMPROVED 20

Kiro now replies in the IDE's configured display language.

— Single sentence with no configuration detail. Editor Actions, Guided Hook Creation, an…

20 **Chat history persists across folder changes** IMPROVED 20

Chat history now persists across workspace folder changes.

— Behavior described but no mechanism or scope given.

Editor Actions, Guided Hook Creation, an…

21 **Agent permission rule improvements** IMPROVED 20

Improves filesystem and command permission rules for agent operations.

— No specifics on which rules changed. Permission Improvements and Configured W…

22 **Ignored files excluded from search** IMPROVED 20

Keeps ignored files out of search results.

— Single line with no scope or config given. Permission Improvements and Configured W…

┗—▷ BREAKING ON UPGRADE

! The IAM Identity Center admin setting 'Kiro Web (Preview)' is renamed to 'Cloud Sessions (Preview)'; organizations that had the prior setting remain enabled, but new organizations must explicitly opt in.

! Cloud sessions default to disabled; organizations that never configured the 'Cloud Sessions' toggle in the Kiro console will find cloud sessions unavailable until an administrator explicitly opts in.

WAS THIS USEFUL? 👍 👎

Anysphere Cursor ⓘ 5 RELEASES · seen 2026-08-19 NOTES ↗

Cursor is an AI-powered code editor built on VS Code that uses large language models to assist with writing, debugging, and refactoring code.

> Cursor shipped a new built-in code hosting platform (Origin) with GitHub sync, PR reviews, and CI app integrations, brought the editor experience to iPad with split-screen review and Apple Pencil markup, tripled Cloud Agent startup speed with pre-warmed Builds, added Google Workspace plugins for Drive, Gmail, and Calendar, and launched a lower-priced Cursor Start plan for India.

└──▷ **WHAT SHIPPED · 13 FEATURES** most completely described first ? what's the number?

01 Builds for Cloud Agents NEW 88

Cloud Agents now boot into pre-prepared Builds — copies of the dev environment with repos cloned, dependencies installed, and the install script already run — instead of setting up from scratch each session, cutting time-to-first-token roughly 3x via environments that boot 10x faster internally. A new `Builds` tab in the Cloud Agents dashboard shows build status, logs, commit SHAs, and which build each agent run used, with `Enable Builds` and `Run setup agent` actions to migrate existing environments, manual build triggering, and agent-driven debugging of failing builds. A configurable staleness threshold controls how often Builds refresh, and if a bad commit or dependency update breaks a Build, agents automatically fall back to the last successful Build while the broken one never becomes active and users are notified; agents can also inspect and manage Builds via built-in tools.

Enable Builds on an existing Cloud Agents environment to get faster agent startup without reconfiguring from scratch.

> 📍 In the Cloud Agents dashboard, open your environment › go to the Builds tab › click 'Enable Builds'. To preview changes first, click 'Run setup agent' instead.

— Names dashboard tab, actions, and fallback mechanism with a usage walkthrough.

`Cloud Agents Start 3x Faster with Builds`

02 Origin code hosting platform launch NEW 80

Origin is a new built-in code hosting platform in Cursor, available in early beta on all paid plans (enterprise org admins can opt out). Repos live under a new `Codebase` tab; clicking `+New` creates a repo whose URL follows the pattern `cursor.com/codebase/<codebase-name>`, and repo creation includes a CLI install flow with commands to clone the repo or push a local project to Origin. Each repo has a per-repo `Settings` page exposing GitHub sync status, access management, and connected apps.

— Names the tab, URL pattern, and settings surface with beta scope. `Origin Code Hosting`

---

03 **Google Workspace plugins: Drive, Gmail, Calendar** NEW

72

A new Google Drive plugin lets agents search files and folders, open and download content, and create and organize files from within Cursor. A new Gmail plugin lets agents search and read mail, draft and send messages, apply labels, and manage threads. A new Google Calendar plugin lets agents read schedules, create and update events, and find free time. All three plugins are browseable and installable from the Cursor Marketplace or the Customize page in Cursor.

— Names each plugin's capabilities and the install surface. `Google Workspace Plugins`

---

04 **Cursor Start plan for India** NEW

70

A new Cursor Start plan is available to developers in India at ₹649/month (tax-inclusive), selectable at `cursor.com/signup` during onboarding or from the dashboard for existing Free users, with UPI and card payment supported for local billing. The plan includes always-on cloud agents that build, test, and ship code continuously in the background, Cursor for iOS with remote control to launch and steer agents from a mobile device, and plugins, MCP servers, hooks, and skills.

— Names price, signup URL, and billing methods but no deeper mechanism. `Cursor Start`

---

05 **App integrations for Origin: Vercel, Depot, Buildkite** NEW

63

Each Origin repo's Apps tab offers app integrations: Vercel for preview deployments per PR, Depot for GitHub Actions CI, and Buildkite for GitHub Actions and native pipelines.

— Names the tab and each integration's specific function. `Origin Code Hosting`

THINNER COVERAGE BELOW

06 **Inbox, SCM providers, and session management on iPad/iOS** NEW

56

A new Inbox view surfaces in-progress work, items needing attention, and PRs currently in review. Bitbucket and Azure DevOps are added as supported SCM providers on iOS/iPadOS. Multi-PR sessions let users open every PR created by a single chat, not just the most recent one, and in-app team switching lets users move between teams they belong to without leaving the app.

— Names each addition and provider though each remains a one-line description.

`Cursor, now on iPad`

---

07 **GitHub sync for Origin repos** NEW

52

GitHub sync lets you connect a GitHub org and pull repos into Origin alongside Cursor-hosted ones; synced repos update in real time, with pushes continuing to GitHub as the source of truth.

— Describes sync direction and source-of-truth behaviour. Origin Code Hosting

08 **Full PR review surface on iPhone and iPad** IMPROVED 50

The mobile PR review surface now covers comments, checks, approvals, reviewer management, and agent-driven resolution — through to merge — on both iPhone and iPad.

— Lists review capabilities covered but no exact UI path given. Cursor, now on iPad

09 **Pull requests in Origin repos** NEW 50

Pull requests on every Origin repo show timeline, commits, checks, diffs, and comments; PRs on synced repos sync bidirectionally with GitHub within seconds.

— Lists PR surface elements and sync timing. Origin Code Hosting

10 **Split-screen PR review mode on iPad** NEW 43

Split-screen mode lets users keep a PR review open alongside a chat session with full file diff rendering.

— Describes the split view and diff rendering but no navigation steps. Cursor, now on iPad

11 **Agents integrated into Origin repos** NEW 42

Agents integrated directly into Origin repos can answer questions about browsed code, make changes, update PRs, or push a branch.

— Names agent actions but no mechanism detail beyond that. Origin Code Hosting

12 **Apple Pencil markup on screenshots** NEW 40

Apple Pencil support enables drawing directly on attached screenshots, and tap-to-place comments let users annotate images at specific points.

— Names the two interactions but no further behavioural detail. Cursor, now on iPad

13 **Cursor launches on iPad** NEW 35

Cursor is now available on iPad for all paid plan subscribers, with a rebuilt layout featuring pinned sidebar chats for monitoring multiple agents simultaneously.

— States availability and layout change with no further mechanism. Cursor, now on iPad

Earendil Works Pi

7 RELEASES · 2026-07-21 → 2026-08-14 NOTES ↗

AI agent toolkit: unified LLM API, agent loop, TUI, coding agent CLI

> Pi shipped a full-featured fullscreen TUI mode (search, selection, scrollback, notifications) alongside a major v4 overhaul of its session and extension APIs, new built-in providers (Baseten, Qwen Token Plan, Kimi Code) and Claude Opus 5 across GitHub Copilot, Anthropic, and Bedrock, plus constrained tool-call sampling, local llama.cpp model management, and new `pi auth` credential commands.

└──▷ WHAT SHIPPED · 31 FEATURES most completely described first ? what's the number?

01 Fullscreen TUI mode and transcript interaction NEW 96

Pi adds a `--tui-mode fullscreen` CLI flag and `/settings` toggle to switch between regular and fullscreen TUI modes at runtime, with a sticky editor/footer dock and independently scrollable transcript, plus a configurable scrollbar (`auto` , `always` , `hidden`) and `scrollbarThumb` theme color. Fullscreen mode gained transcript search via `Ctrl+Shift+F` with incremental highlighting and `Enter` / `Ctrl+G` and `Shift+Enter` / `Ctrl+Shift+G` navigation, a configurable exit-output setting, unbound single-line and half-page scrolling actions, page scrolling and marked-message navigation shortcuts, stacked transient notifications, double-click word selection, granularity-aware drag selection, triple-click paragraph selection, opt-in `Ctrl+P` / `Ctrl+N` prompt history navigation, and the `PI_TUI_ESC_TIMEOUT` environment variable to tune Escape input timeout on high-latency (e.g. SSH) terminals.

Launch the agent in fullscreen TUI mode for a distraction-free session with an independently scrollable transcript.

```
$ pi --tui-mode fullscreen
```

— Names every fullscreen flag, setting, and keybinding shipped. `v0.84.2` `v0.84.1` `v0.84.0`

02 Extension and session API v4 breaking changes BREAKING 92

`ModelsStreamTransforms` is renamed to `ModelsRequestTransforms` ; JSON/RPC `message_update` events now emit only `assistantMessageEvent` deltas, removing the cumulative `message` and `assistantMessageEvent.partial` fields; ModelRegistry.getApiKeyAndHeaders() now returns `ProviderHeaders` with `string | null` values; ModelRegistry.refresh() now accepts `ModelsRefreshOptions` and

returns `ModelsRefreshResult`; ModelRuntime.setRuntimeApiKey() now takes auth cancellation options instead of catalog refresh options (call refresh({ providers: [providerId], signal }) separately); config-form extension OAuth refreshToken(credentials, signal) callbacks must honor a concrete abort signal; dynamic provider refresh store access moves from context.store.read()/context.store.write() to the read-only `context.stored` snapshot and generation-checked context.publish(); the pi-agent-core harness session model is replaced by v4 `Session`, `SessionStorage`, and `SessionRepo` APIs (implement `JsonlSessionRepo` or `InMemorySessionRepo`), with legacy JSONL/in-memory repository APIs removed; the v2 session and `AgentHarness` experimental subpaths are removed and promoted to pi-agent-core's default export; custom `FileSystem` implementations must provide renameFile() with same-filesystem replacement semantics; and `RemoteSession.sessions` no longer exposes runtime phase, model, thinking, attachment, or lock state, now available only via acquired `SessionSnapshot` values through the `SessionMetadata` API.

— Enumerates every renamed/removed API with its migration path. `v0.84.0`

---

03 **Constrained sampling for built-in tools** NEW 87

`Tool.constrainedSampling` adds `prefer` / `require` strict JSON Schema modes plus OpenAI Lark/regex grammar variants, supported across OpenAI, Anthropic, Amazon Bedrock, Google Gemini, and Mistral, gated by new `supportsGrammarTools` and `supportsStrictTools` model capability flags. The experimental `PI_EXPERIMENTAL=1` environment variable enables strict JSON-schema constrained sampling for the default `read`, `bash`, `edit`, and `write` tools.

Enable strict JSON-schema constrained sampling for built-in tools to tighten model output validation during a run.

```
$ PI_EXPERIMENTAL=1 pi
```

— Names the flag, capability checks, and providers supported. `v0.84.2` `v0.82.0`

---

04 **Credential verification and export commands** NEW 86

`pi auth check` verifies provider or model credentials before a run, with optional output of the resolved credential. `pi auth print-api-key` and `pi auth print-bearer-token` export configured credentials to external clients, with automatic OAuth refresh and minimum-validity enforcement.

Verify that your configured provider credentials are valid before starting a long agent run.

```
$ pi auth check
```

Pipe a live-refreshed API key into another tool without manually copying credentials from your config.

```
$ pi auth print-api-key
```

— Three runnable subcommands with example invocations. v0.84.1 v0.83.0

05 **Bash tool session context and streaming RPC events** NEW

84

Built-in and factory-created bash tools now receive `PI_SESSION_ID`, `PI_SESSION_FILE`, `PI_PROVIDER`, `PI_MODEL`, and `PI_REASONING_LEVEL` environment variables so scripts can introspect the active session and model. Direct RPC bash commands also emit streaming `bash_execution_update` RPC events correlated with request IDs for incremental output consumption.

Read session and model context inside a bash tool script to tag output or conditionally branch on the active provider.

```
$ #!/usr/bin/env bash
echo "Session: $PI_SESSION_ID"
echo "Provider: $PI_PROVIDER  Model: $PI_MODEL  Reasoning:
$PI_REASONING_LEVEL"
echo "Session file: $PI_SESSION_FILE"
```

— Five named env vars plus an RPC event, with a runnable example. v0.82.0

06 **Local llama.cpp model management** NEW

81

Pi adds a built-in llama.cpp router with `/login` for connection setup and `/llama` for Hugging Face model search, download, explicit load/unload, and live progress. llama.cpp models now also persist across restarts, staying listed in the catalog before the first successful refresh.

Search for and download a Hugging Face model via the built-in llama.cpp router after connecting to it.

```
$ /login
/llama <model-search-query>
```

— Exact commands (`/login`, `/llama`) and persistence behavior given. v0.82.1 v0.81.0

07 **Custom sampling parameters and vLLM thinking budget** NEW

79

Arbitrary OpenAI-compatible sampling parameters can now be set via `samplingParams` in `models.json`, model overrides, extension providers, and stream options. An opt-in vLLM `thinking_token_budget` setting for OpenAI-compatible models reserves output tokens for the final answer.

Configure custom sampling parameters and a vLLM thinking token budget for a self-hosted OpenAI-compatible model.

json

```
{
  "samplingParams": {
    "temperature": 0.2,
    "top_p": 0.9
  },
  "thinking_token_budget": 2048
}
```

— Config key and file named with a working example. v0.84.0

08 **OAuth and headless login for OpenRouter and Kimi Code** NEW

72

OpenRouter login via `/login` gains an OAuth PKCE flow that mints a user-controlled API key without manual key configuration, plus manual redirect URL and authorization-code entry for remote and headless (SSH) environments where the loopback callback is unavailable. Kimi Code subscription OAuth login for the Kimi For Coding provider is also added via `/login`, including device authorization and automatic token refresh.

— Names the `/login` command and flow variants for each provider. v0.83.0 v0.82.0

09 **ANTHROPIC_AUTH_TOKEN bearer auth for gateways** NEW 69

The `ANTHROPIC_AUTH_TOKEN` environment variable authenticates against Anthropic-compatible gateways using `Authorization: Bearer`, and this auth also covers compaction and branch summary requests.

Authenticate against an Anthropic-compatible gateway (e.g. a corporate proxy) that requires bearer token auth, so compaction and branch summaries also route through it.

```
$ export ANTHROPIC_AUTH_TOKEN=your-token-here
pi
```

— Named env var with a runnable export example. v0.82.1

10 **New built-in providers: Baseten and Qwen Token Plan** NEW

65

Pi adds a built-in Baseten provider authenticated via `BASETEN_API_KEY` with `zai-org/GLM-5.2` as the default model, a Qwen Token Plan Individual provider using the shared international `QWEN_TOKEN_PLAN_API_KEY` environment variable and its documented subscription model catalog, and Qwen Token Plan / Qwen Token Plan China providers with regional endpoints and API-key authentication.

— Names every provider and its auth env var, but no setup command. `v0.84.1` `v0.84.0` `v0.81.0`

---

11 **Experimental remote-session client APIs** NEW 65

Adds transport-neutral `PiClient`, a CBOR protocol, Unix-socket transport, and the `@earendil-works/pi-coding-agent/client` `RemoteSession` controller with transcript reducers.

— Names the client, protocol, and transport but no example. `v0.84.0`

---

12 **get_available_thinking_levels RPC command** NEW 65

The `get_available_thinking_levels` RPC command and RpcClient.getAvailableThinkingLevels() method query supported thinking levels at runtime.

Query which thinking levels are available for the current model via the RPC client in an extension or script.

```javascript
const levels = await rpcClient.getAvailableThinkingLevels();
```

— Named RPC command and method with a code example. `v0.81.0`

---

13 **Claude Opus 5 across GitHub Copilot, Anthropic, and Bedrock** NEW 61

Claude Opus 5 is added through GitHub Copilot with adaptive thinking and a 1M context window, and separately on Anthropic and Amazon Bedrock with adaptive thinking (including an `xhigh` level), inference profiles, and prompt caching.

— Names contexts/features per provider but no config example. `v0.83.0` `v0.82.1`

---

14 **Diagnostics and telemetry improvements** NEW 61

`AssistantMessage.endTurn` preserves OpenAI Codex's terminal `end_turn` signal for diagnostics. Structured Amazon Bedrock failure diagnostics now include HTTP status, modeled error code, and AWS request ID. Vendor-neutral telemetry contracts add agent-owned typed AI-request and harness schemas, composed span starters, and callback helpers. A new `CredentialSynchronizationError` covers credential changes that commit successfully but fail to synchronize local model state.

— Names four distinct diagnostic surfaces, no usage shown. v0.84.2 v0.84.0

15 **TypeBox dependency upgraded to 1.3.7** BREAKING 60

Upgrading TypeBox aliases to 1.3.7 removes deprecated APIs `Type.Base`, `Type.Awaited`, `Type.Promise`, `Type.AsyncIterator`, `Type.Iterator`, `Type.Options`, and `Value.Mutate`; extensions using any of these must migrate to supported TypeBox APIs.

— Names every removed API but gives no replacement mapping. v0.83.0

16 **expandPromptTemplates option for sendUserMessage()** NEW 60

The `expandPromptTemplates` option on extension pi.sendUserMessage() explicitly dispatches commands and expands skills and prompt templates.

Expand prompt templates and skills when programmatically sending a message from a Pi extension.

```javascript
await pi.sendUserMessage('/refactor', { expandPromptTemplates: true });
```

— Named option with a code example showing usage. v0.84.2

THINNER COVERAGE BELOW

17 **--use-theme CLI flag** NEW 58

`--use-theme <name[/name]>` selects a per-run interactive theme without changing saved settings.

Apply a one-off theme for a single session without overwriting your saved theme preference.

```
$ pi --use-theme dracula
```

— Exact flag syntax with a runnable example. v0.84.2

18 **Per-directory AGENTS.override.md context files** NEW 57

Per-directory `AGENTS.override.md` files replace `AGENTS.md` or `CLAUDE.md` in the same directory while preserving context inherited from other directories.

— Names the exact filename and override behavior, no example shown. v0.84.0

19 **Cloudflare AI Gateway binding support** NEW 57

createGatewayBindingFetch() routes Cloudflare AI Gateway requests through a Workers AI binding without needing an API token.

– Names the function and mechanism, no example given. v0.84.2

20 **Streaming stop-reason handling for custom providers** IMPROVED

54

`compat.supportsFinishReason` lets OpenAI-compatible streams that omit `finish_reason` infer normal and tool-use stops when the stream ends. A new `"pending"` stop reason is also added for partial streaming messages in the custom provider stream pattern.

– Names the setting and stop-reason value, no usage example. v0.84.0 v0.83.0

21 **terminate support in blocked tool_call handlers** NEW

50

Blocked extension `tool_call` event handlers gain `terminate` support, letting all-terminating batches skip the automatic follow-up model call.

– Names the mechanism but no code example. v0.84.1

22 **pi.registerMarkdownTransformer() extension hook** NEW

48

Chainable pi.registerMarkdownTransformer() hooks allow display-only transformation of user and assistant Markdown.

– Names the API but no usage example. v0.84.0

23 **defaultTools setting for initial tool selection** NEW

48

The `defaultTools` setting configures the initial built-in tool selection globally or per project.

– Names the setting but not its file location or values. v0.84.2

24 **Extension registration for full pi-ai providers** NEW

45

Extensions can now register complete pi-ai providers, including native authentication, model refresh, filtering, and custom streaming behavior.

– States capabilities exposed but no API name or example. v0.81.0

25 **Model catalog revalidation via If-None-Match** IMPROVED

41

Pi.dev model catalogs now revalidate using `If-None-Match`, so unchanged provider catalogs return an empty `304` response instead of a full download.

— Names the HTTP mechanism but no reader action. v0.82.1

26 **Mermaid and LaTeX rendering in Markdown** NEW 41

Interactive messages gain configurable themed Unicode rendering for Mermaid diagrams, including optional rendering while streaming, plus terminal-friendly Unicode rendering for LaTeX expressions in Markdown.

— Describes what renders but no setting name or example. v0.84.0

27 **Usage accounting for tools, compaction, and branch summaries** IMPROVED 41

Tool, compaction, and branch-summary usage is now persisted and included in session totals, the footer, and session statistics.

— Describes scope but no UI path or field name. v0.81.0

28 **AI_AGENT=pi child-process attribution** NEW 37

`AI_AGENT=pi` is added to CLI and RPC child-process environments for generic agent attribution.

— Names the env var with no further mechanism. v0.84.0

29 **outputPad setting for custom message renderers** NEW 37

The `outputPad` setting is now exposed to custom message renderers in extensions.

— Names the setting with no usage detail. v0.82.1

30 **Exported message and tool lifecycle event types** NEW 32

Message and tool execution lifecycle event types are now exported from the package root.

— States the export but names no specific type. v0.81.0

31 **Inherited per-request fetch injection** NEW 32

Supported text and image provider transports now inherit per-request `fetch` injection.

— Thin one-line mechanism, no example or scope given. v0.83.0

└──▷ **BREAKING ON UPGRADE**

! The `ModelsStreamTransforms` interface is `ModelsRequestTransforms`; extensions referencing `ModelsStreamTransforms` must update to `ModelsRequestTransforms`.

renamed to

! JSON and RPC `message_update` events now emit only `assistantMessageEvent` deltas; the cumulative `message` and `assistantMessageEvent.partial` fields are removed. Clients must assemble deltas between `message_start` and `message_end`.

! ModelRegistry.getApiKeyAndHeaders() now returns `ProviderHeaders` with `string|null` values; extensions that inspect returned headers must handle `null`, and those forwarding headers to pi-ai streams should pass `null` throu unch

! ModelRegistry.refresh() now accepts `ModelsRefreshOptions` and returns `ModelsRefreshResult` instead of discarding cancellation and provider errors.

! ModelRuntime.setRuntimeApiKey() now accepts auth cancellation options instead of catalog refresh options; call refresh({ providers: [providerId], signal }) separately when remote freshness is required.

! Config-form extension OAuth refreshToken(credentials, signal) callbacks must now accept and honor a concrete abort signal.

! Dynamic provider refresh context store access is replaced with the read-only `context.stored` snapshot and generation-checked context.publish() transaction. Handwritten Provider.refreshModels() implementatio must migrate from

! The pi-agent-core harness session model is replaced with the v4 lane-based `Session`, `SessionStorage`, and `SessionRepo` APIs; legacy JSONL and in-memory repository APIs are removed. Use `JsonlSessionRepo` or `InMemorySessionRepo` implementing the new `SessionRepo` contract.

! The v2 session and `AgentHarness` API experimental subpaths are removed; they are now promoted to pi-agent-core's default export.

! Custom harness file-system implementations must now provide FileSystem.renameFile() with same-filesystem replacement semantics.

! `RemoteSession.sessions` no longer exposes runtime phase, model, thinking, attachment, or lock state (previously available as list summaries); that data is now available only from acquired `SessionSnapshot` values via the durable `SessionMetadata` API.

! TypeBox aliases upgraded to 1.3.7 removes deprecated APIs `Type.Base`, `Type.Awaited`, `Type.Promise`, `Type.AsyncIterator`, `Type.Iterator`, `Type.Options`, and `Value.Mutate` — extensions using any of these must migrate to supported TypeBox APIs.

WAS THIS USEFUL? 👍 👎

Charm Crush

4 RELEASES · 2026-07-20 → 2026-08-12 NOTES ↗ CODE ↗

Glamourous agentic coding for all

> Crush's four releases this window are anchored by a new `.crushrc` bash-based configuration system covering providers, models, MCPs, hooks and permissions, alongside OAuth 2.1 authorization for MCP servers, Claude Channels support, new LSP rename/replace tools, and a batch of UI improvements including a scrollable sidebar and session-resume hints.

└──▷ WHAT SHIPPED · 18 FEATURES most completely described first ? what's the number?

01 **.crushrc bash-based configuration file** NEW 95

Adds a `.crushrc` bash-based configuration file, discovered and loaded alongside the existing `crush.json`, enabling shell logic, conditionals, and `source` to include machine-specific overrides (including a local per-directory `.crushrc`). It introduces builtins: `provider add` / `provider remove` (alias `rm`) with `--type` and `--base-url`; `model add` / `model remove` with `--name` and `--context-window`; `mcp add` / `mcp remove` with `--type http`, `--url`, and `--header`; `lsp add` / `lsp remove`; `hook add` / `hook remove`; `permissions allow` / `permissions deny`; and `option reset` to wipe list-type options back to defaults. Scripts can read a `CRUSH_VERSION` environment variable for version-conditional configuration.

Configure a local Ollama provider and model, auto-approve read/edit tools, and add a GitHub MCP server via HTTP — all in a single `.crushrc` file.

bash

```
# ~/.config/crush/.crushrc
provider add ollama --type ollama --base-url
"http://localhost:11434/v1"
model add ollama/llama3.3 --name "Llama 3.3" --context-window
128000
permissions allow view edit
mcp add github \
  --type http \
  --url "https://api.githubcopilot.com/mcp/" \
  --header Authorization "Bearer $GITHUB_TOKEN"
```

Apply machine-specific config conditionally and pull in a separate file kept out of version control.

```bash
# ~/.config/crush/.crushrc
source "$XDG_CONFIG_HOME/squid-config.sh"

if [[ $HOSTNAME == "babysquid" ]]; then
    option skill-path "$HOME/squid-skills"
fi
```

Block a specific tool from ever being offered to the agent, using `permissions deny` in `.crushrc`.

```bash
# ~/.config/crush/.crushrc
permissions deny bash
```

— Full builtin list with flags and runnable config examples `v0.88.0`

02 **Scrollable sidebar with keyboard and mouse navigation** NEW

90

Adds a scrollable sidebar with keyboard focus navigation: `l` `/right` arrow to focus, `j` / `k` to scroll, `g` / `G` for top/bottom, `h` `/left` arrow to return to chat, `tab` to return to editor, plus mouse-wheel scrolling support when the sidebar is focused.

— Every keybinding named, directly usable `v0.86.0`

03 **OAuth 2.1 authorization for MCP servers** NEW

80

On a 401, Crush automatically runs a browser-based PKCE authorization-code flow for HTTP MCP servers, persisting tokens (mode `0600`) to `~/.config/crush/mcp-oauth-tokens.json`; headless/SSH environments instead receive a printed authorization URL. SSE transport also gains OAuth support, with metadata fixups and resource parameter stripping.

— Names exact token path, permission mode and transport details `v0.87.0`

04 **Claude Channels MCP extension support** NEW

80

Adds `--channels <server>:<name>` persistent flag, working on both `crush` and `crush run`, to opt individual sessions into Claude Channels MCP extension servers, enabling MCPs to push notifications into a running Crush session.

Opt a session into a Claude Channels-capable MCP server so it can push notifications (e.g. CI alerts) into your Crush session.

```
$ crush run --channels server:webhook 'Watch for build failures and summarize them'
```

— Exact flag syntax with a runnable command example v0.87.0

05 **Stats subcommand aggregation flags** NEW 80

Adds `--all` flag to the `stats` subcommand to aggregate usage stats from all known projects (sourced from `projects.json`), and a `--crawl-dir` flag to discover and aggregate stats from older projects in a directory tree that predate the `projects.json` registry.

Aggregate token/cost stats across every Crush project tracked in your install — useful for a weekly spend review.

```
$ crush stats --all
```

Discover and aggregate stats from older projects in a directory tree that predate the projects.json registry.

```
$ crush stats --crawl-dir ~/src
```

— Both flags shown with runnable example commands v0.86.0

06 **LSP rename and replace tools** NEW 75

Adds `lsp_rename` to rename a symbol and all its references across the project, and `lsp_replace_symbol` to replace, insert before/after, or delete an entire function, method, or class by name. Also adds fancy diff view rendering for LSP tool operations.

— Named tools with behavior, no direct invocation syntax shown v0.86.0

07 **OAuth dialog URL copy keybinding** NEW 70

Adds `u` keybinding in the OAuth dialog to copy the verification URL to the clipboard, enabling authentication over SSH where the browser cannot open automatically.

— Exact key and use case, no further mechanism v0.86.0

08 **Session-resume hint on exit** NEW 65

Displays a splash screen on exit with the session ID and the `crush -s <id>` command needed to resume the session.

— Exact resume command given, minimal mechanism detail v0.89.0

09 **ctrl+end chat follow keybind** NEW 65

Adds `ctrl+end` keybind to jump to the bottom of the chat and re-enable auto-follow as new messages stream in.

— Exact keybind and behavior, no further mechanism v0.89.0

THINNER COVERAGE BELOW

10 **Bash tool command display improvements** IMPROVED 55

Adds bash syntax highlighting to the display of commands Crush executes via the Bash tool, improving legibility of long shell invocations, and strips redundant `cd -to-project` prefixes from that display to reduce visual noise.

— Two named display tweaks, no configuration surface v0.89.0

11 **System-wide configuration file** NEW 50

Crush now loads `/etc/crush/crush.json` as a site-wide config file, improving NixOS and multi-user compatibility.

— Exact path named, but thin on mechanism v0.87.0

12 **LSP performance optimization** IMPROVED 45

Improves LSP performance via pre-`$PATH` server filtering, significantly reducing CPU and memory usage in large repositories.

— Mechanism named, no user-facing control v0.87.0

13 **OAuth dialog and quit dialog hints** NEW 45

Adds a hint in the quit dialog that pressing `ctrl+c` twice skips the confirmation entirely.

— Exact keypress named, minor UI hint v0.86.0

14 **Bang-mode output streaming performance** IMPROVED 40

Improves bang-mode output streaming performance, eliminating quadratic rendering cost for shell command output.

— Mechanism named, no operational surface v0.87.0

15 **AWS Bedrock SSO token-refresh** IMPROVED 40

Crush detects expired AWS Bedrock SSO tokens, presents a re-authentication prompt, and automatically retries the interrupted turn.

— Behavior explained, no manual command exposed v0.88.0

16 **Attachment chip removal button** NEW 40

Adds a clickable `X` button on attachment chips to remove attachments with the mouse.

— Small UI addition, clear but minor v0.86.0

17 **Recovery from mid-stream provider connection resets** IMPROVED 35

Adds graceful recovery from mid-stream provider connection resets so interrupted responses restart cleanly.

— Behavior described only, no mechanism or control v0.86.0

18 **Client-server connection resilience** IMPROVED 30

Adds automatic event-stream reconnection in client mode when the connection to a Crush server drops, and a UI distinction between a lost server connection and an uninitialized agent.

— Two thin behavior notes, no user action v0.87.0

└──▷ ALSO FROM THESE RELEASES

WAS THIS USEFUL? 👍 👎

Block Goose

3 RELEASES · 2026-07-23 → 2026-08-12 NOTES ↗ CODE ↗

an open source, extensible AI agent that goes beyond code suggestions - install, execute, edit, and test with any LLM

> Goose shipped three fast-moving releases adding a hooks system with PreToolUse tool-call denial, 25+ new LLM provider integrations, a cluster of new slash commands, and a wide set of environment-variable controls, alongside desktop UI navigation, localization, and OAuth improvements — while removing the CLI `goose project` subcommand.

└──▷ **WHAT SHIPPED · 34 FEATURES** most completely described first ? what's the number?

01

New environment-variable configuration options NEW

85

Adds `LOCAL_WHISPER_LANGUAGE` to configure local Whisper transcription language, `GOOSE_OAUTH_CALLBACK_PORT` to pin a stable OAuth `redirect_uri` and avoid port-collision re-auth failures, `GOOSE_MAX_TOOL_RESPONSE_SIZE` to cap tool output size returned to the model, `GOOSE_FAST_MODEL` (consumed by `ModelConfig::with_fast`) to designate a lightweight model for fast subtasks, `MAX_CODE_BLOCK_LINES` to control TUI code-block truncation, and `GOOSE_DOCS_ROOT` to point Goose at a local docs mirror for air-gapped deployments.

Pin the OAuth callback port so the redirect_uri stays stable across restarts and avoids re-authentication prompts.

```
$ export GOOSE_OAUTH_CALLBACK_PORT=9876
goose session
```

Cap how much tool output is fed back to the model to avoid blowing context limits on noisy tools.

```
$ export GOOSE_MAX_TOOL_RESPONSE_SIZE=65536
goose session
```

Host Goose documentation locally for a network-isolated environment so the agent resolves docs without hitting the internet.

```
$ export GOOSE_DOCS_ROOT=/mnt/internal-docs/goose
goose session start
```

— Names six env vars with runnable examples for three of them `v1.46.0` `v1.45.0`

02

Task scheduler: `--enable-scheduler` flag and quarterly option

NEW

80

Adds `--enable-scheduler` flag to both the `serve` and `acp` subcommands to opt in to scheduled recipe execution (disabled by default for ACP), and adds a `quarterly` scheduling option to the task scheduler.

Enable scheduled recipe execution when running the Goose ACP server, which is off by default in this release.

```
$ goose serve --enable-scheduler
```

— Exact flag, subcommands and runnable example given v1.45.0 v1.46.0

03 **25+ new LLM provider integrations** NEW 75

Adds new providers Celeris, Friendli, Azure AI Foundry (multi-LLM), OllamaCloud (dynamic model discovery and automatic context-limit detection), Sakana AI (OpenAI-compatible Fugu API), iFlytek Spark, Astron MaaS, Fireworks AI, OpenRouter (with request parameters), Together AI, OrcaRouter, Perplexity, Alibaba/Qwen via DashScope, Databricks AI Gateway, Scaleway, NEAR AI Cloud, EmpirioLabs, xAI SuperGrok (OAuth subscription), Vercel AI Gateway, Routstr, FuturMix, oMLX, Atomic Chat, Muse Spark 1.1 (Meta Models API), and AWS Bedrock for OpenAI GPT models via a mantle endpoint.

— Enumerates every provider but no selection command shown v1.46.0 v1.44.0

04 **Session editing before fork via `--edit`** NEW 75

Adds `--edit` flag to the `session` command to edit a conversation before forking it into a new session.

Edit a previous session's conversation before branching it into a new fork — useful for removing sensitive context or reframing a task before sharing.

```
$ goose session --edit <session-id>
```

— Exact flag and runnable example command provided v1.46.0 v1.44.0

05 **New slash commands and model switching** NEW 70

Adds `/goal` so the agent self-evaluates against stated objectives before finishing a session, `/status` for a live session status summary, and `/model` to switch the LLM mid-session without restarting; the `model` command also gained tab completion and provider switching.

— Named commands with clear purpose, no invocation example v1.46.0 v1.45.0

06 **Hooks system with PreToolUse tool-call denial** NEW 70

Adds a hooks system with `PreToolUse` denial capability, letting operators block specific tool invocations before they execute; the Stop hook context also gains a `working_dir` field so hooks can inspect the working directory at session end.

— Explains mechanism, no example hook configuration given v1.46.0

07 **Global agent hints from `~/.agents/AGENTS.md`** NEW 70

Loads global agent hints from `~/.agents/AGENTS.md` automatically at session start.

— Names exact file path, no content format shown v1.46.0

08 **`goose://new-session` and `goose://resume` deep links** NEW

65

Supports `goose://new-session` and `goose://resume` deep links, including an initial prompt passed via `goose://new-session`.

— Named URI scheme, no full example URL given `v1.46.0`

---

09 **Summon extension peek mode for delegate tasks** NEW

65

Adds Summon extension peek mode for async background delegate tasks, plus `context` and `working_dir` parameters for delegate tasks.

— Names parameters, no example invocation shown `v1.46.0`

---

10 **OAuth and authentication improvements** NEW

65

Adds Hugging Face OAuth support with a dedicated auth tab in settings; adds `logo_uri` to OAuth client ID metadata documents so authorization servers can display the goose logo on consent screens; adds proactive OAuth token refresh to avoid re-authentication at every session start; and adds Azure AD (Entra ID) bearer token authentication via the `AZURE_OPENAI_AD_TOKEN` environment variable.

— Names field and env var, no setup walkthrough `v1.46.0`

---

11 **CLI `goose project` subcommand removed** BREAKING

65

The `goose project` subcommand (and its `p` alias) and the `goose projects` subcommand have been removed; CLI project support is no longer available.

— Names exact removed commands and alias `v1.46.0`

---

12 **Hidden background sessions via ACP `session/new`** NEW

60

Adds `_meta.hidden` field to the `session/new` ACP endpoint to create hidden (background) sessions.

— Names exact field and endpoint, no full request example `v1.46.0`

THINNER COVERAGE BELOW

13 **Desktop navigation panel and session browsing improvements** NEW

55

Adds a resizable sidebar with a drag handle whose width is persisted across sessions; displays session metadata (name, date, etc.) on sidebar chat-item hover; adds chat history search in the navigation panel; groups chat sessions by project with collapsible project group headers; adds an interactive menu for single-select elicitations in the UI; adds a search/filter field to the provider selection grid; and adds delete support for custom apps from the Apps UI.

— Lists UI additions, no exact navigation paths given v1.46.0 v1.44.0

14 **Per-message token, cost and latency stats** NEW

55

Adds per-message token count, cost, TTFT, and tok/s stats to the UI alongside derived session totals, including cache-token tracking for accurate cost reporting.

— Explains metrics shown, no UI location named v1.46.0 v1.44.0

15 **Local inference platform support: MLX, Vulkan, Termux** NEW

55

Adds MLX model support to the local inference provider, adds Linux Vulkan support for local inference, and adds Termux detection in the installer to automatically select the musl portable build.

— Names platforms, no build or install command v1.46.0

16 **Desktop UI localization for 15 languages** NEW

55

Adds desktop locales for French, German, Italian, Portuguese, Indonesian, Malay, Vietnamese, zh-TW, Korean, Spanish, Japanese, Hindi, Russian, Turkish, and Simplified Chinese (zh-CN), plus in-app language selection.

— Lists all languages, no menu path given v1.46.0 v1.44.0

17 **`goose review` subcommand for local code review** NEW

50

Adds `goose review` subcommand for local AI-powered code review.

— Names subcommand but gives no usage detail v1.46.0

18 **TLS support for `acp serve`** NEW

50

Adds TLS support to the `acp serve` command for encrypted ACP server deployments.

— Named command, no certificate or flag config shown v1.46.0

19 **TUI diff viewer and live shell streaming** NEW

50

Adds a TUI diff viewer for reviewing file changes inside the terminal interface, and streams shell command output in real time as commands run rather than buffering until completion.

— Describes behaviour, no navigation path given v1.46.0

20 **MCP extension and plugin support** IMPROVED 50

Adds MCP extensions support in open plugins and the ability to bind MCP apps to trusted ownership metadata, and upgrades the MCP transport layer to rmcp 2.0.

— Names transport version, no plugin config example v1.46.0 v1.45.0

21 **Skills listing and selective disabling** NEW 50

Adds a CLI command to list loaded skills with their token counts, and adds opt-in ability to disable built-in skills individually.

— Describes capability, no exact command name given v1.46.0

22 **Cross-language SDK bindings** NEW 50

Adds UniFFI SDK for cross-language bindings, Python wheel publishing for provider bindings, and Kotlin FFI with Maven publish.

— Names build artifacts, no install command v1.46.0

23 **Image handling in agent tools** NEW 50

Adds an image read tool so the agent can inspect image files directly, and forwards images and MCP embedded-resource blobs in Anthropic and Google provider formats.

— Names providers, no tool invocation example v1.46.0

24 **New model catalog additions** NEW 45

Adds known models GPT-5.6, GPT-5.5, GLM-5.2, Opus5 (with adaptive thinking), claude-sonnet-5, claude-fable-5, Cursor composer-2.5, Cursor-agent (fetched from CLI), latest Gemini models including Gemini 3.x, MiniMax-M3 and M2.7, and OVHcloud Qwen3.6-27B.

— Names model identifiers but no selection mechanism v1.46.0 v1.45.0 v1.44.0

25 **GenAI semantic convention attributes in telemetry** IMPROVED 45

Adds GenAI semantic convention attributes to OpenTelemetry spans and enriches the root span with `gen_ai` attributes.

— Names attribute namespace, no span example shown v1.46.0

26 **Encrypted Nostr-based session sharing** NEW 40

Adds encrypted Nostr-based session sharing and session import.

— Thin description, no protocol or command detail v1.46.0

27 **Overlapping-window command classifier chunking** IMPROVED 40

Chunks command-classifier input with overlapping windows to improve security classification coverage.

— Explains mechanism, no threshold or config given v1.44.0

28 **Tool-call label enrichment gated by ACP capability** IMPROVED 40

Gates tool-call label enrichment on ACP client capability, avoiding enrichment overhead for clients that do not advertise support.

— Explains condition, no capability flag name given v1.45.0

29 **Markdown export format for sessions** NEW 35

Adds Markdown as an export format option for session export, joining existing formats.

— Names format only, no export command shown v1.46.0

30 **Projects as backend session sources** NEW 35

Adds projects as backend session sources with system prompt injection.

— Thin description, no configuration shown v1.46.0

31 **Automatic desktop ACP session reconnection** IMPROVED 30

Reconnects desktop ACP sessions automatically after sleep or connection loss.

— States behaviour, no toggle or config named v1.46.0 v1.44.0

32 **Stable agent event message identity** IMPROVED 30

Introduces stable agent event message identity for consistent event tracking across the agent loop.

— Describes concept, no field name given v1.45.0

33 **Harbor eval runner for agent evaluation** NEW 25

Adds Harbor eval runner for automated agent evaluation.

— Bare mention with no usage detail v1.46.0

34 **Unified thinking-effort control across providers** NEW 25

Adds unified thinking-effort control across all providers.

— Bare mention, no parameter name given `v1.46.0`

└──▷ BREAKING ON UPGRADE

! The `goose projec t` subcommand (and its `p` alias) and the `goose project s` subcommand have been removed; CLI project support is no longer available.

WAS THIS USEFUL? 👍 👎

Google gemini-cli ⓘ 2 RELEASES · 2026-07-28 → 2026-08-11 NOTES ↗

An open-source AI agent that brings the power of Gemini directly into your terminal.

> This window's headline items are two large internal maintainer-automation suites — a caretaker agent for automated GitHub issue triage/evaluation and a PR-generator agent for automated bug-fixing — alongside smaller CLI additions: tool registry discovery, an eval coverage report command, and a Vertex AI base URL update.

└──▷ **WHAT SHIPPED · 5 FEATURES** most completely described first ? what's the number?

01 Caretaker agent for automated issue triage and evaluation NEW

74

Ships a caretaker agent suite: a Cloud Run webhook ingestion service for automated intake of GitHub events, a re-triage workflow with issue comment handling so new comments trigger fresh triage runs, an LLM-based triage orchestrator (`caretaker-triage`) with prompt hill-climbing and container build support for automated GitHub issue classification, a triage evaluation framework and judge runner (`caretaker-evals`), local golden issue collection and Firestore sync tools, a Cloud Run job entrypoint for running caretaker evals at scale, publication of a workable-spec event to a ready-for-code Pub/Sub topic when triage completes, a GCP deployment script for caretaker agent services, and updates to the caretaker Firestore schema adding `error` and `pr_number` fields.

— Extensive named surfaces but internal infra with no user-facing invocation shown v0.55.1

v0.53.0

02 PR-generator agent for automated bug-fixing NEW

74

Introduces a PR-generator orchestrator with an iterative bug-fixing state machine and container worker entrypoint, an Antigravity agent runner and prompt templates, a Firestore dual-locking concurrency mechanism and test ingestion utilities for the PR-generator database, an environment config parser, command executor, and GitHub integration for the PR-generator core, plus Cloud Run job, Workflows definition, and Dockerfile configuration for PR-generator infrastructure.

— Detailed mechanism and named components but no runnable command for readers v0.55.1

THINNER COVERAGE BELOW

03 **Eval coverage report command** NEW 40

Adds an eval coverage report command (`feat(evals): add eval coverage report command`) in the `evals` package for measuring evaluation test coverage.

— Names the package and commit message but no exact CLI invocation v0.53.0 v0.55.1

04 **Tool registry discovery** NEW 34

The CLI can now automatically find and enumerate available tools at runtime via a new tool registry discovery mechanism.

— Names the capability but no flag, command, or config to invoke it v0.55.1

05 **Vertex AI base URL updated** IMPROVED 23

Updates the Vertex AI base URL, keeping Vertex AI auth paths current.

— Bare mention of a URL change with no detail on scope or impact v0.55.1

WAS THIS USEFUL? 👍 👎

◆ AI Agent Frameworks

NVIDIA Object-Oriented Agents (NOOA) ⓘ

2 RELEASES · 2026-07-30 → 2026-08-18 NOTES ↗

NVIDIA Object Oriented Agents: the Pythonic way to build AI Agents.

> Object-Oriented Agents shipped runtime LLM selection via a callable @strategy(llm=...), an extensible `nooa` CLI, portable trace journaling, and CyberGym benchmark agents, alongside security hardening across the viewer API, playground model configuration, and `ShellTools`, plus a breaking rename of all `nemo_flow_*` symbols to `nemo_relay_*`.

└──▷ WHAT SHIPPED · 10 FEATURES most completely described first ? what's the number?

01 Runtime-selectable LLM via callable @strategy IMPROVED

88

The @strategy(llm=...) decorator now accepts a callable in addition to a static client, letting agents choose which LLM backend to use at call time rather than binding it at class-definition time, e.g. @strategy(llm=pick_llm) where pick_llm() returns a client via `nooa.unifiedllm.get_llm_client`.

Dynamically select an LLM per-call (e.g. based on task complexity) instead of binding one model at class definition time.

python

```python
from nooa import Agent
from nooa.unifiedllm import get_llm_client

def pick_llm():
    return get_llm_client("ollama_chat/qwen3:1.7b",
api_base="http://localhost:11434")

class AdaptiveAgent(Agent):
    @strategy(llm=pick_llm)
    async def analyze(self, text: str) -> str:
        """Analyze the text."""
        ...
```

— Runnable code example with exact decorator syntax. v0.0.9

02 nemo_flow_* renamed to nemo_relay_* BREAKING

65

All `nemo_flow_*` symbols are renamed to `nemo_relay_*`; any code importing or referencing `nemo_flow_*` names will break and must be updated to the new prefix.

— Exact renamed symbol prefixes given, no migration guide. v0.0.9

03 **Playground custom-model endpoint restriction** IMPROVED

65

Playground custom-model configuration now constrains the `endpoint` and `api_key_env` fields to server-declared pairs, blocking arbitrary endpoint injection.

— Names exact fields restricted, no example call. v0.0.7

04 **Entry-point extensibility for nooa CLI** NEW

60

External packages can now register new `nooa` subcommands via Python entry points, making the `nooa` CLI extensible without modifying core.

— Names the entry-point mechanism but gives no example. v0.0.9

THINNER COVERAGE BELOW

05 **Viewer API authentication and resource limits** IMPROVED

55

The viewer API now requires an authentication token and enforces CORS restrictions to prevent unauthenticated access to trace data, and enforces resource limits on viewer ingest to bound memory and CPU consumption under high trace volume.

— Names auth/CORS/resource-limit mechanisms, no config keys. v0.0.7

06 **ShellTools directory confinement** IMPROVED

50

`ShellTools` file operations are now confined within the current working directory, reducing the blast radius of LLM-generated shell actions.

— Names ShellTools but no flag or config surface. v0.0.7

07 **Portable journal exporter for tracing** NEW

45

Adds a portable journal file exporter for tracing, enabling offline and cross-environment trace storage so traces can be moved between environments.

— No named flag, format, or file path given. v0.0.9

08 **CyberGym benchmark agent examples** NEW

45

Adds a NOOA agent example for the CyberGym benchmark environment demonstrating how to build a cyber-domain agent with the OO Agents framework, later extended with a portfolio-based NOOA agent variant.

— Names benchmark but no code or config specifics. v0.0.9 v0.0.7

09 **LLM routing metadata in observability traces** IMPROVED

30

Adds LLM routing metadata to generation observability, surfacing which backend handled each LLM call in traces.

— Brief description, no field name or example. v0.0.9

10 **Secret redaction in telemetry** NEW

20

v0.0.7 adds secret redaction in telemetry, intended to prevent sensitive values from appearing in trace or telemetry data.

— Only named in summary, no mechanism or config given. v0.0.7

└──▷ **BREAKING ON UPGRADE**

! All `nemo_flow_*` symbols are renamed to `nemo_relay_*`; any code importing or referencing `nemo_flow_*` names will break.

WAS THIS USEFUL? 👍 👎

Nous Research Hermes

6 RELEASES · 2026-07-20 → 2026-08-18

NOTES ↗ CODE ↗

The agent that grows with you

> Across six releases Hermes added streaming conversational voice with barge-in, a desktop plugin SDK, Agent-to-Agent (A2A v1.0) protocol support, signed outbound webhooks, a grounded-citations research/fact-checking skill, a smart-approvals overhaul, and a wave of new CLI commands, model providers, and multi-machine Bot Mode chat features.

└──▷ WHAT SHIPPED · 45 FEATURES most completely described first ? what's the number?

01 `hermes sessions export` with redaction NEW

95

Adds `hermes sessions export` with output formats Markdown, Quarto, HTML, prompt-only, and Hugging Face-ready traces; filter flags for age, workspace, and platform; an `--redact` secret-scrubbing pass; and compacted-session lineage stitching.

Export a workspace's sessions as a Hugging Face-ready trace dataset with secrets scrubbed, for use as fine-tuning or eval data.

```
$ hermes sessions export --redact
```

— Runnable command with named formats, flags and example. v2026.7.20

02 Terminal backend and provider status APIs NEW

95

Adds `GET /api/tools/terminal/backends` and `PUT /api/tools/terminal/backend` REST endpoints to enumerate and set the terminal execution backend (local / docker / ssh / modal / daytona / singularity), paired with a terminal execution backend picker in the Desktop Capabilities tab showing per-backend health probes (docker CLI+daemon, ssh config keys, modal/daytona credentials, singularity/apptainer binary) and setup guidance. Also adds a `GET /api/tools/toolsets/{name}/config` per-provider `status` field (`ready` / `needs_keys` / `needs_auth` / `needs_setup`) so the tab shows truthful provider readiness instead of a guessed 'Ready' pill.

Check which terminal execution backends are available and their health status before switching to Docker isolation.

```
$ curl -s http://localhost:<port>/api/tools/terminal/backends |
jq .
```

— Names exact REST endpoints with a runnable health-check example. v2026.7.20

03 **`key_cmd` dynamic credential source for providers** NEW

95

Adds `key_cmd` credential source in the `providers` config block, letting Hermes run an arbitrary command to mint short-lived bearer tokens instead of storing static keys — output can be a bare token or JSON with `access_token` / `expires_in` fields; tokens are cached until near expiry.

Use a short-lived SSO token from an internal auth CLI instead of a static API key, so credentials never go stale mid-session.

yaml

```yaml
providers:
  my-gateway:
    base_url: https://gateway.internal.example.com/v1
    api_mode: chat_completions
    key_cmd: my-auth-cli print-token --profile prod
```

— Exact config key with a runnable YAML example. v2026.8.16

04 **Desktop plugin SDK: rendering, storage, and APIs** NEW

90

Adds a plugin SDK to the desktop app, with Kanban as the founding plugin, `ctx.download` for delivering files to users, floating pane placement, and multiple GUI windows (r4). Exports `McpTab`, `ToolsetConfigPanel`, and `HermesGateway` type from `@hermes/plugin-sdk`, plus host.getGateway() returning the live `$gateway` instance, so desktop plugins can render full credential/OAuth/MCP configuration panels identical to Settings → Capabilities (r2). Enables plugins to render inline components inside assistant messages via `::name{...}` directives, including `::preview` for live in-message page previews, and adds per-plugin durable data directories that survive plugin updates and removal (r0). Adds `profiles.list` and `profiles.create` ws JSON-RPC methods (in `tui_gateway/methods_profiles.py`) for enumerating and creating profiles, plus an `image.generate` ws RPC for plugin surfaces to trigger image generation through the agent backend (r3).

Enumerate all profiles from a desktop plugin to build a roster UI, including a last-session preview per profile.

```javascript
// Inside a desktop plugin, via host.request ws JSON-RPC
host.request('profiles.list', { include_sessions: true })
```

— Enumerates every named export, directive, and RPC across four releases. v2026.8.18 v2026.8.16 v2026.8.13 v2026.8.3

05 **New CLI utility commands** NEW 90

Adds `!command` shell-escape syntax to run a shell command instantly without consuming a model turn, `/init` to scan a project and generate or update an `AGENTS.md` file, `/diff` to show staged, all, or session file changes from any surface, `/context` to break down what is filling the current context window, and `/focus` for a reduced-output view with hidden-line recovery. Also adds `hermes import-agent` to migrate a Claude Code or Codex CLI setup into Hermes in one command, and `hermes approvals suggest` to mine approval history into allowlist proposals.

Run a shell command from within a Hermes CLI session without spending a model turn — useful for quick checks like listing files or running a test while staying in context.

```
$ !ls -la
```

Migrate an existing Claude Code or Codex CLI configuration into Hermes in one step.

```
$ hermes import-agent
```

— Every command named with runnable examples for two of them. v2026.8.3

06 **Smart approvals: default review, deny rules, and circuit breaker**

IMPROVED 80

Enables smart approvals (LLM-based per-command review) as the new default instead of prompting the user for every flagged command, adds user-defined deny rules that block commands even under yolo mode, and adds a `/deny <reason>` command so the agent receives an explanation when a command is refused and can course-correct (r5). Extends this with a consecutive-denial circuit breaker that stops misbehaving approval-request loops, plus a new approval gate for docker/podman daemon-redirect commands (r4).

— Names the command and behaviour change but no deny-rule config file. v2026.8.3 v2026.7.20

07 **Pluggable SecretSource for Bitwarden and 1Password** NEW

80

Adds a pluggable `SecretSource` interface supporting Bitwarden and 1Password (`op://` references) as secret backends, with multiple vaults enabled simultaneously, deterministic precedence, conflict warnings, and per-variable provenance.

— Names interface and reference syntax, no CLI example. v2026.7.20

08 **Cold-start and startup latency reductions** IMPROVED 80

Reduces `hermes -w` cold-start time from ~14s to ~1.8s and makes `hermes update` no-ops 2–6s faster, adding prompt caching for tool schemas on native Anthropic (r4). Separately achieves ~80% reduction in cold-start first-turn latency (approximately 4.3s to 0.9s) across CLI, gateway, TUI, desktop, and cron by moving Discord capability detection off the critical path with a token-keyed 24-hour disk cache and background refresh (r5).

— Precise before/after metrics for two separate optimizations. v2026.8.3 v2026.7.20

09 **`/subscription` and `/topup` billing commands** NEW 75

Adds `/subscription` and `/topup` commands to the TUI and CLI for managing Nous plans — view allowance, preview upgrade cost, apply changes, and undo — without leaving the terminal; desktop gets a matching billing settings tab.

— Named commands and desktop counterpart, no worked example. v2026.7.20

10 **Signed outbound webhooks for lifecycle events** NEW 70

Adds signed outbound webhooks that push HMAC-signed lifecycle events (session activity, turn completions, tool events) to any registered HTTP endpoint, enabling integration without polling.

— Names signing scheme and event types, no endpoint registration syntax. v2026.8.3

11 **Context compaction and pruning for long sessions** NEW

70

Adds proactive tool-result pruning, per-turn micro-compaction, a guaranteed N-user-message tail, and configurable compression thresholds per-model and in absolute tokens to keep long sessions coherent.

— Names mechanisms and thresholds, no exact config keys given. v2026.8.3

12 **Bot Mode plugin and multi-machine bot chat** NEW 70

Bundles Bot Mode (`hermes-bots`) as a built-in, default-on desktop plugin implementing the core teammate protocol for Bot Chat sessions, syncing with a multi-source roster (r1). Adds cross-connection Bot Mode with a Create-on picker for launching multi-machine group chats, plus a disband (delete) action for group chats (r0).

Adds capability-refresh and timeless prompts to keep Bot Chat sessions eternal, and a one-time protocol upgrade path for legacy Bot Chat sessions (r1). The desktop app also gains multi-source agent support spanning sockets, roster, SDK, and fan-out updates (r2).

— Names plugin and protocol across three releases, no exact command. v2026.8.18 v2026.8.16.2 v2026.8.16

13 **New model providers and model support** NEW 70

Bundles the Meta Model API (Muse Spark) provider plugin and adds Meta Muse Spark 1.2 to the OpenRouter curated picker (r0). Adds Fireworks AI (with cost estimation and a #2 slot in the provider picker) and DeepInfra as first-class providers, plus Upstage Solar; adds support for GPT-5.6 (Sol/Terra/Luna + Pro variants), grok-4.5 (GA), moonshotai/kimi-k3, claude-fable-5, claude-sonnet-5, tencent/hy3 (GA), and LM Studio JIT model loading for local setups (r5).

— Names every provider and model, no config switch shown. v2026.8.18 v2026.7.20

14 **Live-viewable subagent transcripts** NEW 70

Adds live-viewable subagent transcripts: `delegate_task` dispatches return tail-able transcript files with every tool call, result, and streamed reply per child agent.

Watch a subagent work in real time by tailing its live transcript file the moment it is dispatched via delegate_task.

```
$ tail -f <transcript-file>
```

— Names the mechanism with a runnable tail command. v2026.7.20

15 **Remote gateway headers and self-healing connections** NEW 70

Adds `remote gateway headers` support in the desktop app, carried through the connections registry, test probes, and Settings UI. Adds status bar reconnect for offline gateways and self-healing for dropped SSH/HTTP registered remote connections, plus connection-aware plugin routing that scopes plugin socket connections and session/pin lists per connection.

— Names UI surfaces and registry, no exact header config shown. v2026.8.16.2

16 **Grounded citations skill and fact-checking mode** NEW 65

Adds the `grounded-citations` skill, making Hermes produce research where every claim is backed by a verifiable source — quotes are matched against actual page text and citations link to exact evidence. The same skill powers a fact-checking mode that

evaluates any document or claim and reports what checks out, what does not, and what could not be verified.

— Names the skill and mechanism but no exact invocation shown. v2026.8.3

17 **Streaming conversational voice with barge-in** NEW 65

Adds streaming, clause-by-clause conversational voice with barge-in (interrupt mid-sentence by speaking), busy-aware silence detection, and hands-free control across the CLI, desktop, and gateway adapters.

— Describes mechanism and surfaces but no config example. v2026.8.3

18 **Voice note transcription and auto-TTS on messaging platforms** NEW 65

Adds voice note transcription and auto-TTS replies on WhatsApp, Feishu, DingTalk, LINE, QQ, Photon, and Weixin, with platform-aware codec selection (opus where required, correct caption attachment).

— Names every platform and codec detail, no config steps. v2026.8.3

19 **NVIDIA SkillEvaluator advisory scanning for skill installs** NEW 65

Adds NVIDIA SkillEvaluator Tier 1 advisory scanning (license and security checks) on skill installs, plus project-skill quarantine and non-interactive trust inheritance.

— Names scanner tier and checks, no CLI flag shown. v2026.8.18

20 **Durable delivery-obligation ledger for channel replies** NEW 65

Adds a durable delivery-obligation ledger in `state.db` that records final responses around the platform send and redelivers them on next boot, closing a silent-loss window for Telegram, Discord, Slack, and other channels.

— Names storage file and channels, no direct user action. v2026.7.20

21 **Profile-based message routing on multiplexed gateway** NEW 65

Adds profile-based message routing to the multiplexed gateway, letting a single bot token route specific guilds, channels, or threads to isolated profiles each with their own config, skills, memory, and secrets.

— Explains routing mechanism, no config example shown. v2026.7.20

22 **`hermes peer` bot-to-bot direct messages** NEW 65

Adds `hermes peer` CLI subcommand for bot-to-bot DMs across machines and gateways.

Send a direct message from one bot to another across machines or gateways without opening an interactive session.

```
$ hermes peer <target> "Run the nightly audit and report back"
```

— Runnable command example provided. v2026.8.18

23 **`max` and `ultra` reasoning effort tiers** NEW 65

Adds `max` and `ultra` reasoning effort tiers for GPT-5.6 and Codex, per-model reasoning-effort overrides in config, per-slot effort in MoA presets, and per-task effort for auxiliary models.

— Names tiers and config surfaces, no exact keys. v2026.7.20

24 **Incremental markdown rendering performance** IMPROVED 65

Adds incremental markdown rendering to the TUI so output paints per token instead of waiting for complete lines, and reduces desktop streaming markdown CPU cost by 14× via incremental block lexing, virtualizing review-pane diffs to eliminate Shiki freeze on large outputs.

— Concrete 14× metric named, no toggle exposed. v2026.7.20

25 **Artifact rendering in desktop app** NEW 60

Adds artifact rendering in the desktop app: versioned cards with sandboxed live preview in a right-rail viewer so generated HTML/apps run safely alongside the chat.

— Names viewer location and sandboxing, no API given. v2026.8.3

26 **Cua Driver 0.20 support for computer-use** NEW 60

Supports Cua Driver 0.20 runtime contracts for computer-use, including auto-repair of installed drivers that fail the runtime contract at update or runtime (r1), and adds a user-facing authorization flow for cua-driver browser attachment in computer-use mode (r2).

— Names driver version and auto-repair, no manual command given. v2026.8.16.2

v2026.8.16

27 **Codex OAuth context raised to 900K tokens** IMPROVED 60

Raises Codex OAuth context to 900K tokens for the `gpt-5.6` family and `gpt-5.4` (subscription 1M rollout), up from a previous raise to 350K for live-verified sessions.

— Exact token limits and models named, no user action needed. v2026.8.16.2

28 **STT as first-class `hermes tools` category** NEW 60

Adds STT as its own `hermes tools` category with GUI toggles, dashboard dropdowns, unified language resolution, and support for OpenAI's gpt-transcribe.

— Names category and provider, no exact command shown. v2026.8.3

29 **Declarative memory provider panel** NEW 60

Adds a declarative memory provider panel with a full-config modal in Desktop settings, with provider config schemas in `plugins/memory/*/config_schema.py`.

— Names config schema path, no exact UI steps given. v2026.7.20

30 **MCP server setup: consent card and composer suggestions** NEW 60

Adds a `setup_mcp` tool with an inline MCP consent card rendered as an interactive clarify-style blocking bridge in the desktop transcript, letting users approve MCP server connections without leaving the conversation, and suggests MCP servers from the composer draft as brand pills in the desktop UI.

— Names the tool and UI pattern, no CLI equivalent given. v2026.8.13

THINNER COVERAGE BELOW

31 **Agent-to-Agent (A2A v1.0) protocol support** NEW 55

Adds Agent-to-Agent (A2A v1.0) protocol support as a bundled plugin, allowing Hermes to discover, communicate with, and be driven by other A2A-compatible agents.

— Names protocol version and plugin, no setup steps. v2026.8.3

32 **On-device wake word detection and multi-profile voice routing** NEW 55

Adds on-device open-vocabulary wake word detection and multi-profile voice routing, so different wake phrases can reach different profiles without audio leaving the device.

— Clear mechanism, no setup steps given. v2026.8.3

33 **Cron workflow enhancements in desktop** IMPROVED 55

Adds cron and blueprint recipe shortcuts to the sidebar nav rail and surfaces model drift impact in the desktop cron view (r3), and adds a per-job model picker to the cron create/edit dialog (r5).

— Names UI locations across two releases, no config keys. v2026.8.13 v2026.7.20

34 **Mid-turn redirect during active turns** NEW 55

Adds mid-turn redirect capability: type a correction while the agent is working and the active turn course-corrects with the new guidance, preserving work in flight and the original prompt.

— Explains mechanism and behavior, no exact command shown. v2026.8.3

35 **Tool-calling iteration limit raised to 500** IMPROVED 50

Raises the default tool-calling iteration limit from 90 to 500, removing an artificial wall for long autonomous runs.

— Exact before/after numbers but no config key to adjust it. v2026.8.3

36 **Composer suggestion and clarify UX improvements** IMPROVED 50

Adds skill-match, connection-repair, and recurrence-to-cron suggestion providers in the desktop composer, labels the agent's recommended choice on every clarify surface (rendered in tertiary text), and quietly suppresses composer suggestions the user has repeatedly ignored.

— Lists suggestion types, no way to configure them noted. v2026.8.13

37 **`context-manager` protocol on SessionDB** NEW 45

Adds `context-manager` protocol support on `SessionDB` for safer session database lifecycle management.

— Names protocol and target class, no usage steps. v2026.8.16.2

38 **Per-route toolset overrides for webhook agent runs** NEW 40

Adds per-route toolset overrides for webhook agent runs, allowing different tool sets to be configured per webhook route.

— States capability without naming the route config key. v2026.8.13

39 **Desktop TTS/STT settings surfacing** IMPROVED 40

Surfaces all xAI TTS parameters in the desktop GUI config, and lists config-defined command TTS/STT providers in desktop settings.

— Thin description of two settings additions. v2026.7.20

40 **Turn-path observability in Desktop/TUI** NEW 40

Adds per-turn wall-clock duration display in the desktop transcript, and `INFO` log records for 'prompt accepted / turn finished' on the Desktop/TUI turn path.

— Names log messages and UI element, minimal further detail. v2026.8.16

41 **Build/runtime mismatch warning in About** NEW 40

Surfaces an in-app warning (with installer link) in About when the app build is out of sync with the runtime.

— Names exact UI location, thin otherwise. v2026.8.18

42 **Global-hotkey quick-entry window** NEW 35

Adds a global-hotkey quick-entry window that captures a thought into any session from anywhere in the OS.

— Describes function but no hotkey or config named. v2026.8.3

43 **Widget clicks as agentic response path** NEW 35

Widget clicks now reach the agent as hidden user turns, making widget interaction a first-class agentic response path.

— Describes behavior only, no configuration surface. v2026.8.18

44 **Browser Use mode benchmark and CLI 3.0 mode** NEW 30

Adds Browser Use CLI 3.0 mode, and adds the Browser Use mode A/B benchmark to the evals suite (from PR #81958).

— Named PR reference but little detail on the mode itself. v2026.8.18

45 **Marquee animation for clipped row titles** NEW 25

Adds marquee animation for clipped inline row titles on hover in the desktop app.

— Minor cosmetic change with no further detail. v2026.8.13

WAS THIS USEFUL? 👍 👎

CrewAI

10 RELEASES · 2026-07-20 → 2026-08-14 NOTES ↗

Framework for orchestrating role-playing, autonomous AI agents. By fostering collaborative intelligence, CrewAI empowers agents to work together seamlessly, tackling complex tasks.

> CrewAI's window focused on deeper flow and deployment observability (execution UUIDs, exception recording, outcome/duration/HITL signals, FlowFailedEvent), a more trustworthy skills system, three new built-in tools, and a unified `crewai create` scaffolding command with enterprise project-ID linking.

└──▷ WHAT SHIPPED · 8 FEATURES most completely described first ? what's the number?

01 New built-in tools: URLReadTool, WaitTool, IBM Db2 search NEW

80

The tool library gains `URLReadTool` for fetching and reading content from arbitrary URLs inside agent workflows, `WaitTool` for pausing agent execution while waiting on long-running background jobs, and an IBM Db2 search tool extending the library's built-in data-source integrations.

Give an agent the ability to retrieve live web content by attaching URLReadTool to its tool list.

python

```python
from crewai_tools import URLReadTool

url_tool = URLReadTool()

agent = Agent(
    role='Web Researcher',
    goal='Gather information from public URLs',
    backstory='An expert at reading web content',
    tools=[url_tool]
)
```

— Names three tools with a runnable usage example for one 1.15.12 1.15.11 1.15.8

02 Flow and deployment run telemetry NEW

65

Flow execution now carries execution context management with UUID support for tracing individual flow runs, records the type of exception that ended a flow for richer post-mortem observability in AMP traces, tracks when a trace batch is shared with AMP, and counts deployments from any origin while recording where each deployment started. Flow execution also now reports outcome, duration, and human-in-the-loop signals as part of flow telemetry, and CrewAI emits a `FlowFailedEvent` when a flow execution fails, enabling event-driven error handling in Flows.

— Names several telemetry signals but no user-facing API to query them 1.15.16 1.15.15 1.15.9

03 **Unified `crewai create` scaffolding command** IMPROVED

65

Project scaffolding is unified under the `crewai create <resource>` subcommand, replacing the prior separate scaffold commands.

— Exact command given but no full usage walkthrough 1.15.12

THINNER COVERAGE BELOW

04 **Project ID tracking for enterprise linkage** NEW

57

Runtime context is now separated from the coding agent, allowing project ID to be tracked independently per execution, and CrewAI adds a `project_id` field to link open-source CrewAI usage to an enterprise account.

— Names the `project_id` field but no usage example given 1.15.14 1.15.11

05 **Skills system: disclosure, usage events, registry auth** NEW

53

Skills now support progressive disclosure, revealing skill detail incrementally rather than all at once, and CrewAI emits skill usage events at runtime so observability pipelines can capture when and how skills are invoked by agents. Downloads from the skill registry now support authentication, enabling access to protected or private skills.

— Three distinct skill-system additions named but no commands shown 1.15.9 1.15.7 1.15.7a1 1.15.5

06 **Accurate tool failure reporting** IMPROVED

40

Tool failures are now surfaced as actual failures instead of being silently reported as success, giving agents accurate tool execution feedback.

— Clear before/after but no named surface or user action 1.15.9

07 **Coding agent visibility in telemetry** NEW

38

CrewAI AMP is now surfaced in `AGENTS.md`, telemetry adds coding-agent detection, and interception-hook dispatches are tracked in telemetry for improved observability of agent workflows.

— Names AGENTS.md and hook tracking but mechanism is thin 1.15.11

08 **App metadata on platform action tools** NEW

25

Platform action tools now carry app metadata, surfacing richer context for actions executed via the CrewAI platform.

— Single thin line with no mechanism or example 1.15.12

WAS THIS USEFUL? 👍 👎

LangChain LangGraph

4 RELEASES · 2026-07-28 → 2026-08-11 NOTES ↗

Build resilient agents.

> LangGraph's window centers on new per-node tracing controls via `add_node`'s `trace_policy` parameter (alongside a breaking removal of `tags` from `TracePolicy`) and a new `omit_expired` flag for skipping stale rows in checkpoint reads.

└──▷ WHAT SHIPPED · 4 FEATURES most completely described first ? what's the number?

01 **Per-node trace_policy parameter on add_node** NEW 67

`add_node` now accepts a `trace_policy` parameter, letting callers set per-node tracing behavior directly when wiring the graph, e.g. graph.add_node('my_node', my_node_fn, trace_policy=TracePolicy(...)).

Attach a per-node trace policy at graph construction time to control which nodes are traced in production.

```python
graph.add_node('my_node', my_node_fn,
trace_policy=TracePolicy(...))
```

– Names the exact parameter and shows a runnable code example `1.2.11` `1.2.10`

02 **omit_expired flag for checkpoint reads** NEW 65

Adds an opt-in `omit_expired` parameter to checkpoint and checkpoint-postgres readers, letting callers skip expired rows when reading checkpoint history to reduce noise and improve read performance in long-running workflows.

– Names exact flag and both packages but no code example `checkpoint==4.2.0` `checkpointpostgres==3.1.1`

THINNER COVERAGE BELOW

03 **Tags field removed from TracePolicy** BREAKING 55

The `tags` field has been removed from `TracePolicy`, narrowing the tracing configuration surface; any code that sets `tags` on a `TracePolicy` instance will break.

– Names the removed field but gives no migration path `1.2.10`

04 **Typed stream_events return and native projections** NEW 30

Adds typed return for v3 `stream_events` and native projections support.

— One-line mention with no mechanism or usage shown `1.2.10`

└──▷ BREAKING ON UPGRADE

! The `tags` field has been removed from `TracePolicy`; any code that sets `tags` on a `TracePolicy` instance will break.

WAS THIS USEFUL? 👍 👎

holmesgpt

5 RELEASES · 2026-07-27 → 2026-08-10 NOTES ↗

SRE Agent - CNCF Sandbox Project

> HolmesGPT's five releases in this window centered on human-in-the-loop safety controls, new MCP integrations (Atlassian Rovo, multi-org GitHub Apps), and a ~50-60% cut in prompt token usage, alongside smaller auth, logging and observability additions.

└──▷ **WHAT SHIPPED · 11 FEATURES** most completely described first ? what's the number?

01 **Pre-acquired AZURE_AD_TOKEN support for Entra ID auth** NEW

75

Supports the pre-acquired `AZURE_AD_TOKEN` environment variable for Entra ID authentication, enabling token-based Azure auth flows without interactive login.

— Names the exact env var and the auth flow it enables `0.37.0`

02 **OOM remedy logging via TOOL_MEMORY_LIMIT_MB** IMPROVED

65

Logs the `TOOL_MEMORY_LIMIT_MB` remedy when a tool is OOM killed, making memory-limit guidance visible in the investigation output.

— Names the exact environment variable a reader can set `0.38.1`

THINNER COVERAGE BELOW

03 **RFC 8707 resource indicator support in MCP OAuth** IMPROVED

55

Adds RFC 8707 resource indicator support to the MCP OAuth flow, enabling resource-aware token requests during MCP authentication.

— Names the RFC and flow but no flag or config shown `0.38.2`

04 **Atlassian Rovo MCP integration** NEW

50

Adds Atlassian Rovo MCP integration, enabling access to Jira issues and Confluence pages via Atlassian's hosted MCP server.

— Names the integration and data sources, lacks setup detail `0.39.0`

05 **Prompt and tool description size reduction** IMPROVED

45

Reduces system prompt and tool description sizes by ~50-60%, significantly cutting prompt token consumption per investigation.

— Quantified improvement but no way to invoke or verify it `0.38.0`

06 **GitHub App multi-organization support in MCP addon** IMPROVED

45

Extends the GitHub MCP addon to support GitHub App installations spanning multiple organizations.

— Names the addon and scope but no setup steps given `0.39.0`

07 **Remote tool call approval workflows** NEW

40

Implements remote tool call approval workflows, enabling human-in-the-loop gating of agent-initiated tool executions.

— Describes the mechanism but no config or command shown `0.38.0`

08 **Conversation-worker capacity visibility and reconnect bounding** IMPROVED

40

Makes conversation-worker slot exhaustion visible and bounds reconnect sign-in attempts, improving observability of capacity limits.

— Describes the change without a metric, flag or config `0.38.0`

09 **JSON logging format for log scrapers** NEW

40

Adds JSON logging format support for log scrapers, enabling structured log ingestion pipelines.

— Names the format but no config key or flag given `0.37.0`

10 **Langfuse support in OpenTelemetry tracing** NEW

40

Adds Langfuse support in OpenTelemetry tracing for LLM observability and trace analysis.

— Names the integration but no setup steps provided `0.37.0`

11 **Slack channel-history context recovery eval coverage** NEW

20

Adds Slack channel-history context recovery evaluation coverage.

— Bare mention with no detail on scope or mechanism `0.38.0`

WAS THIS USEFUL? 👍 👎

OpenClaw

1 RELEASE · 2026-08-08 NOTES ↗ CODE ↗

Your own personal AI assistant. Any OS.

> OpenClaw v2026.6.33 focuses on hardening agent authority across MCP sessions, gateway actions, and tool allowlists, while adding extended-stable package channel support and fixing message delivery reliability in Discord and Telegram integrations.

└──▷ WHAT SHIPPED · 7 FEATURES most completely described first ? what's the number?

01 **Extended-stable package channel support** NEW 65

Package installations can now select, update from, and receive availability notices for the `extended-stable` channel, without silently falling back to another release line.

— Names the exact channel; no install command shown v2026.6.33

THINNER COVERAGE BELOW

02 **Session-bound grants for MCP loopback clients** IMPROVED 58

External MCP loopback clients now use short-lived, session-bound attach grants instead of inheriting mutable child-process authority, which limits lateral movement from a compromised plugin.

— Mechanism explained but no exact config or flag named v2026.6.33

03 **Trusted provenance enforcement for gateway message actions** IMPROVED 45

Gateway message actions now retain trusted requester provenance and reject untrusted callers, closing an escalation path that previously existed through the action bridge.

— Describes the fix but no operable surface named v2026.6.33

04 **Improved liveness and watchdog stall detection** IMPROVED 45

Liveness checks and watchdog semantics now distinguish genuine agent stalls from active long model calls and wedged backends, reducing false-positive run terminations.

— Explains behavior change but no configurable parameter named v2026.6.33

05 **Factory-owned tool allowlists** IMPROVED 40

Narrow tool allowlists remain owned by the factory that constructs them, preventing downstream components from widening their own privilege set.

— States the safeguard but not how to invoke or configure it v2026.6.33

06 **Reliable Discord reconnect message delivery** IMPROVED

40

Discord reconnects no longer silently drop queued messages or repeat ambiguous non-idempotent sends.

— Clear fix description but no reproduction or config detail v2026.6.33

07 **Telegram bot-to-bot and reply-fence handling** IMPROVED

40

Telegram bot-to-bot and reply-fence handling now preserves intended thread context and authorization results on delivery.

— Describes fix but no concrete setting or example v2026.6.33

WAS THIS USEFUL? 👍 👎

◆ AI Model & Data Infrastructure

emisar

11 RELEASES · 2026-07-20 → 2026-08-18 NOTES ↗

An MCP that lets AI tools securely connect to your infrastructure, write IaaS code, debug issues, and assist during incidents - without risking production stability. Built for security teams to approve and infrastructure teams to experience like magic.

> Emisar's biggest work this window was a runbook overhaul — typed, staged execution plans with a slug-keyed MCP draft/publish workflow and immutable per-run snapshots — alongside tighter runner and pack scope enforcement, trust-gated dispatch, self-updating runners via `emisar update`, and steady catalog growth to 100 packs and 1,689 actions.

└──▷ WHAT SHIPPED · 39 FEATURES most completely described first ? what's the number?

01 **Runner lifecycle flags for install.sh** NEW 86

Adds `--uninstall` flag to `install.sh` to remove the cached token, legacy token file, and generated runner identity while preserving configuration, local evidence, and logs. Adds `--reset-identity` for unattended runner identity resets when supplying a different enrollment key during reinstall, and `--purge` to remove all retained files including those preserved by `--uninstall`. Interactive reinstall with a different enrollment key now prompts whether to preserve the existing external identity or reset it.

Cleanly decommission a runner — wiping its cached token and generated identity — without losing its config, logs, or local evidence.

```
$ install.sh --uninstall
```

Force an unattended identity reset when re-enrolling a runner with a different enrollment key, for use in automation or CI pipelines.

```
$ install.sh --reset-identity
```

— Exact flags with runnable examples v0.33.0

02 **Runbook draft and versioning workflow via MCP** BREAKING 83

Adds `get_runbook` and `update_runbook_draft` MCP endpoints keyed by slug, letting agents replace the single unpublished draft under the exact hash they read while

keeping publication human-only — agents can revise and test a runbook draft before a human publishes it. Publish confirmation now renders a diff of changed lines against the canonical text whose hash defines the release. Running an older (non-live) release now returns `not_live` instead of silently dispatching current content; running an unpublished draft requires explicit consent plus the hash of the draft as read. Each execution snapshots the definition it dispatched, making the audit record immutable to later edits, publishes, or deletes. The Run button in the history list names the live release directly (e.g., `Run v3`) and marks a waiting unpublished change with a visual indicator. A portal migration collapsed each runbook's per-save version rows into a single runbook record and renumbered published versions into releases, running automatically before the instance serves traffic on upgrade — existing version numbers change.

— Names exact endpoints, statuses, and migration behavior `v0.39.0` `v0.37.0`

03 **Self-updating runners via `emisar update`** NEW 80

Adds `emisar update` subcommand for installer-managed runners to self-update, with release checksum and GitHub build attestation verification before executing stop, swap, restart, and rollback.

Update an installer-managed runner in place — the runner validates the release checksum and GitHub build attestation before applying the stop/swap/restart transaction.

```
$ emisar update
```

— Runnable command with verification steps named `v0.40.0`

04 **Streaming run output via wait_for_run** IMPROVED 78

Adds `wait_for_run` MCP call that accepts an output cursor and returns the next one, reading the event log forward; it fragments oversized events, wakes on new progress, and can page a finished run's trimmed output back to the caller. When a terminal runbook result exceeds one MCP response, `wait_for_run` now returns the summary in the first response and an opaque continuation token for ordered 64 KiB pages, removing the previous arbitrary total-step ceiling.

— Names endpoint, page size, and removed limit `v0.40.0` `v0.34.0`

05 **Runbook authoring safeguards** IMPROVED 73

Policy overrides that cannot match any action ID — such as regex-style globs like `cassandra\.drop_*` — are now flagged with a warning while you write them, surfacing deny rules that silently protect nothing. Runbook projection failures now report the actual size rather than returning a not-found response, so oversized runbooks are no longer silently dropped from `list_runbooks` or denied in `get_runbook`. Character limits on runbook title and description now carry byte bounds derived from the projection budget, preventing multibyte-encoded descriptions (e.g. Japanese or

accented Latin) at the documented 4,096-character limit from causing runbooks to vanish.

— Names exact endpoints and the byte limit `v0.38.0`

---

06 **Environment variable runner relabeling** NEW 73

Adds `EMISAR_GROUP` and `EMISAR_RUNNER_ID` environment variables to relabel a fleet runner without editing per-host configs.

Relabel all runners in a fleet to a named group and assign unique IDs without touching per-host config files.

```
$ EMISAR_GROUP=prod-eu EMISAR_RUNNER_ID=runner-42 ./emisar-runner
```

— Exact env vars with runnable example `v0.37.0`

---

07 **Role-aware console with billing-manager and scoped-admin controls**

NEW 70

Billing managers now get a finance-only console, while operators can own agents, approvals, and runbooks without gaining team or policy administration rights. Scoped admins are blocked from delegating more reach than they hold or arming account-wide pack cleanup. Reads that expose every pack version or an exact run command now require a checked subject rather than relying on already-filtered callers, and the read-only staff console now requires MFA proof tied to the current enrollment. Console navigation, actions, filters, and empty states adapt to the member's role and access — billing managers no longer see a dead Dashboard link, missing runner access is surfaced as a permission state, and restricted pack and action views explain why results are limited.

— Explains mechanism and scope but no exact command `v0.41.0`

---

08 **Runbook staged execution model with typed inputs** NEW

70

Runbooks now support typed inputs bound at declaration time, with stages that run sequentially or in parallel against selected runner groups, and steps that extract named outputs, test success conditions, and wait within explicit bounds. A single approval freezes the complete runbook execution plan before any work begins, and execution pages show attempts, outputs, waits, and terminal causes in order. Canonical JSON for runbook definitions uses the same shape across the console and MCP, enabling programmatic construction and review of execution plans.

— Thorough mechanism, lacks a named endpoint `v0.36.0`

---

09 **Runner and pack scope enforcement across console and MCP**

IMPROVED

67

A member is now scoped to only the runners and groups they are permitted to use. Runner, pack, action, approval, and audit discovery enforces the member's current runner and pack scope across both the console and MCP; catalog rows require deployment on a runner the member can see, and malformed cross-account associations fail closed. MCP catalog, search, exact lookup, runner inspection, and runbook recovery now exclude pending, rejected, revoked, retired, hash-mismatched, and incomplete pack versions from model-visible results, and the Packs page follows the live catalog, granting unadvertised versions one day to disappear before cleanup.

— Names affected surfaces but no direct command `v0.40.0` `v0.33.0` `v0.32.0`

---

10 **`no_new_privs` enforcement for action children**

BREAKING

67

Action children now start with `no_new_privs`, preventing `execve` inside a pack from gaining setuid or file-capability privileges the runner does not already hold. For example, `postqueue` (setgid `postdrop`) can no longer grant access to the Postfix queue — the runner user must own the queue directly or be a `postdrop` member.

— Concrete before/after example, no migration command `v0.37.0`

---

11 **Pack catalog growth and new integrations** NEW

65

The catalog expanded steadily: 91 packs/1,386 actions added bounded diagnostics for GCP, Pure FlashArray, Terraform, Nomad, OIDC/JWKS, nftables, TCP, and Docker Compose; 95 packs/1,498 actions added Apache Airflow, Spark, Google Cloud billing, and BunnyCDN with production MIG rollout placement in available zones; 100 packs/1,682 actions added JFrog Artifactory, Databricks, Sentry, Symbolicator, and NTPsec plus expanded Cassandra and Cloudflare coverage; 100 packs/1,689 actions added Consul registration-churn snapshots, debugging-pack process context/environment-key/argv/connection reads, GCP Monitoring Cloud Logging name and entry reads, and HCP Terraform plan summaries exposing replacement paths; a further release expanded the infrastructure diagnostics catalog with additional checks; and a runner release added BunnyCDN operations.

— Enumerates counts and integrations, no usage shown `v0.41.0` `v0.40.0` `v0.37.0` `runner-v0.17.2` `v0.36.0` `mcp-v0.5.0`

---

12 **`--json` flag for `emisar pack update`** NEW

65

Adds `--json` flag to `emisar pack update` that emits a partial machine-readable report before returning a post-update validation error, preserving automation output on failure.

Capture machine-readable pack update output even when post-update validation fails, so CI pipelines can parse the partial report before acting on the error.

```
$ emisar pack update --json
```

— Exact flag and failure behavior given `v0.40.0`

13 **run_action evidence and expected fields** NEW 65

Extends `run_action` to accept optional `evidence` and `expected` fields alongside `reason` (now up to 2000 characters), letting an agent carry its full reasoning chain; the approval screen and run details render evidence, expected outcome, and reason together for reviewers.

— Names exact fields and limit, no example call `v0.32.0`

14 **Multi-arch runner container image with provenance** NEW 65

Publishes an official multi-architecture container image at `ghcr.io/andrewdryga/emisar-runner` with build provenance and an SBOM.

Verify the integrity and provenance of a downloaded runner binary and the official container image before deploying.

```
$ gh attestation verify emisar-0.18.0-linux-amd64.tar.gz --owner
andrewdryga
sha256sum -c SHA256SUMS
gh attestation verify oci://ghcr.io/andrewdryga/emisar-
runner:0.18.0 --owner andrewdryga
```

— Names image path with verification example `v0.37.0`

15 **Trust-gated action dispatch and revalidation** IMPROVED 62

Dispatch revalidates the action contract at execution time so a stale page or tool call cannot execute after trust changes, and the runner UI locks the Run button for unavailable actions with an explanation. Runbook recovery fails closed if trust changes between action inspection and execution, preventing hidden pack version disclosure. Pack-trust conflict messages now name the specific runners that disagree about an action instead of failing generically, and pre-run dispatch rejections (contract changes, refusals, rate limits) now log bounded, allowlisted fields so rejected calls are visible in operations.

— Mechanism detailed, no exact endpoint given `v0.34.0` `v0.33.0`

16 **`list_runners` MCP endpoint** NEW 62

Adds `list_runners` to the MCP API, inlining each runner's dispatchable pack IDs so a single call reveals what a named host can do.

— Names endpoint and behavior, no example call v0.32.0

17 **Real-agent conformance evals** NEW 62

Introduces real-agent conformance evals: a scheduled workflow drives the live Claude Code and Codex CLIs through a fail-closed loopback relay against a live stack, hard-failing on policy-blocked calls, invalid mutation arguments, a `run_action` without a prior `get_action` for the same action, placeholder reasons, and runs not driven to a terminal status.

— Detailed mechanism, no user-facing action v0.32.0

THINNER COVERAGE BELOW

18 **Secret redaction expansion in runner v0.19.0** IMPROVED 58

Runner `v0.19.0` expands automatic secret redaction to cover connection strings, database URLs, key-derivation inputs (salt, pepper), cookie and session signing keys, and passphrase-pattern field names — acting as a safety net when actions omit their own redaction declarations.

— Names exact field categories covered, no toggle v0.39.0

19 **Runner token auto-rotation** IMPROVED 53

Runner tokens now carry a 90-day bounded lifetime and self-refresh two-thirds of the way through over the existing connection, with expired tokens refused at connect.

— Exact lifetime and refresh timing, no config key v0.37.0

20 **Immediate session termination on scope and admin changes** IMPROVED 50

Role, runner, and pack scope changes now take effect in open web sessions immediately, reconnecting the session before stale authority can be acted on. Ending a member's sessions now disconnects the live console session immediately, not only the cookie, closing the window where an active console remained usable after an administrator ended sessions mid-incident.

— Clear mechanism, no config surface named v0.41.0 v0.38.0

21 **pfSense pack secure reads** IMPROVED 50

pfSense pack gains resolver, NTP, and WireGuard peer reads that never expose private keys, plus a DHCP reservation action staged for operator approval. pfSense certificate

reads no longer emit the certificate's private key.

— Names pack and reads but no config keys `v0.39.0`

---

22 **Pack behavior test harness** IMPROVED 50

Behavior harness for packs now runs against a real service manager booted as PID 1, a per-case Docker daemon, iptables inside its namespace, and a real dpkg database for install, remove, and autoremove scenarios; uncovered cases are recorded with an explicit reason.

— Describes test infra, not user-facing action `v0.39.0`

---

23 **Pack execution safety constraints for curl, jq, and output size** IMPROVED 48

Every pack's `curl` is confined to an explicit protocol with globbing disabled, blocking URL expansion or credential exfiltration from API-supplied URLs. Each pack's structured output is bounded to fit the runner's cap at its advertised worst case, and jq filters are restricted to core builtins for compatibility with minimal hosts.

— Names curl and jq constraints, no exact limits `v0.37.0`

---

24 **Runbook targeting for runner groups** NEW 47

Runbook targets can now name a runner group in the model contract, with explicit runbook targets supported, and runbook target selection in the console scales to large fleets with a stable trigger, a searchable dense roster, and cardinality-encoding scope icons.

— Describes UI and contract change, no exact syntax `v0.37.0` `runner-v0.17.2`

---

25 **Copy-ready pagination for MCP reads and SSO group mappings** IMPROVED 47

Adds paginated MCP reads that return a copy-ready `next` call (echoing one object) instead of a bare cursor, so agents can continue pagination without parsing a raw cursor value. SSO group mappings now paginate in both directions.

— Describes mechanism, no endpoint names given `v0.41.0` `v0.32.0`

26 **Pack action failure handling on missing commands and HTTP errors**

IMPROVED

45

Missing source commands and HTTP error responses now fail pack actions instead of silently passing through an empty success via downstream pipes or successful transport. Curl-backed API actions now fail on 4xx and 5xx responses instead of reporting transport success.

– Clear behavior change, no config surface named v0.36.0 v0.34.0

27 **AI client conformance for ChatGPT and Claude** IMPROVED

43

ChatGPT tool annotations now distinguish read-only calls from mutations, and domain verification accepts OpenAI's text challenge. Claude conformance evals select MCP authentication by mode and skip interactive permission prompts during headless runs.

– Names two AI clients, lacks operational detail v0.33.0

28 **MCP protocol version adoption** IMPROVED

43

The MCP bridge and portal adopt the 2026-07-28 MCP routing headers and dual-era endpoint, including OAuth Client ID Metadata Documents.

– Names protocol date and OAuth mechanism, no steps v0.36.0

29 **MCP dispatch signing hardening** IMPROVED

43

MCP signed dispatch now signs and verifies the narrative a human approver actually reads (attestation v5) rather than a reconstruction of it. The MCP bridge signing key is now sourced from a pinned credential directory instead of the environment.

– Names attestation version, no path or key format v0.37.0

30 **Production incident response skill** NEW

40

Adds a new public `respond-to-production-incidents` skill giving customer agents a bounded observe, diagnose, act, and verify workflow.

– Names the skill but not its invocation v0.33.0

31 **MFA enrollment safeguards and step-up rate limiting** IMPROVED

40

MFA enrollment and recovery-code regeneration both require proof of the current inbox; credential step-up codes are now rate-limited cluster-wide rather than per node.

– States mechanism, lacks limit numbers v0.37.0

32 **Sensitive read approval gating and masking** IMPROVED

40

Credential-returning reads are approval-gated rather than classified low risk; sensitive run values are masked in a single pass so one match cannot rewrite another's marker.

— Describes mechanism, no named endpoint v0.37.0

33 **Typed JSON action results** NEW 35

Enables actions to opt into typed JSON results, dispatched against the pinned trusted descriptor.

— Thin description, no format details v0.32.0

34 **Registry catalog CDN compaction** IMPROVED 35

Serves the registry catalog compact and gzip-encoded behind a CDN, reducing transfer to roughly one-tenth of the previous size.

— Gives a size figure but no config surface v0.32.0

35 **Enrollment key removed from process command line** IMPROVED 35

The fleet installer no longer places the reusable enrollment key on the process command line.

— Names the fix, no replacement mechanism given v0.39.0

36 **Runner connection protocol efficiency improvements** IMPROVED 30

Backlogged run output now ships as one frame instead of one per line, and the connection lease renews at half its life instead of on every heartbeat.

— States change without measured impact v0.37.0

37 **Device grant audit logging** NEW 30

Device grant claiming writes an audit row per minted key, naming the approver.

— Brief mechanism, no query or endpoint given v0.34.0

38 **Console operator input persistence** IMPROVED 30

Console operator input (approval notes, grant scope selections, policy overrides) now survives re-renders caused by co-approver broadcasts or refused submits.

— Describes fix without reproduction steps v0.34.0

39 **llms.txt website index** NEW 20

Adds an `llms.txt` index to the website.

— Single-line addition with no further detail v0.34.0

└──▷ BREAKING ON UPGRADE

! The portal migration collapses each runbook's per-save version rows into a single runbook record, renumbers published versions into releases, and repoints all execution history — existing version numbers will change. The migration runs automatically before the instance serves traffic on upgrade.

! Action children now run with `no_new_privs`, so any setuid or setgid helper in a runner's process tree no longer elevates. For example, `postqueue` (setgid `postdrop`) can no longer grant access to the Postfix queue — the runner user must own the queue directly or be a `postdrop` member.

WAS THIS USEFUL? 👍 👎

Hugging Face

6 RELEASES · seen 2026-08-19 API ↗

Hugging Face is a platform providing pre-trained machine learning models, datasets, and tools for natural language processing and computer vision tasks.

> Hugging Face published a full public API spanning 316 endpoints across 27 areas, and shipped six product updates including AI-agent-driven Space creation, Jobs label filtering, granular resource-group permissions, egress usage dashboards, and MCP Server upgrades with a unified `hf_fs` tool and secure Sandboxes.

└──▷ WHAT SHIPPED · 7 FEATURES most completely described first ? what's the number?

01 **Unified hf_fs tool in MCP Server** NEW 70

The Hugging Face MCP Server adds `hf_fs`, a single interface covering repositories, storage, documentation, and papers with built-in search, operable in just over 1,000 tokens.

— Names the tool and its token budget precisely MCP Server Enhancements

02 **AI agent-driven Space creation** NEW 65

A new option on the Space creation page at `huggingface.co/new-space` lets users build a Space using an AI agent: the page generates a command to paste into your agent, which then builds and iterates on the Space automatically. The builder supports targeting a model, paper, or local folder as the source.

— Exact UI entry point and generated-command workflow given Build Spaces with AI Agents

03 **Filter Jobs pages by label** NEW 65

User and organization Jobs pages now support label-based filtering, with the most-used labels surfaced as clickable chips showing job counts, plus free-form `key=value` label input to filter by any label including those not shown as chips.

— Names the exact filter syntax and UI mechanism Filter Jobs by Label

04 **Public API launch across 27 areas** NEW 60

Hugging Face published a full public API (openapi.json) totaling 316 endpoints across 27 areas, including Spaces (41 endpoints), Datasets (36), Models (36), SCIM (28, for managing Enterprise organization member access — requires organization owner and a write-permission access token), Jobs (23), Discussions (20), Collections (18), and Organizations (16), plus 19 more areas: Buckets, Users, Agentic Provisioning, Resource groups, Service Accounts, Paper pages, Webhooks, Notifications, Kernels, OAuth, Documentation, Repositories, SQL Console, Inference Endpoints, Repository Search, Agents, Auth, Container Registry, and Tokens.

— Names every area and endpoint count but no method/path specifics `0.0.1`

05 **Sandboxes for MCP Server assistants** NEW 60

The MCP Server adds Sandboxes, giving AI assistants secure execution environments attached to buckets and repositories for tasks like dataset analysis, model training, and Space creation.

— Describes mechanism and use cases but no invocation detail `MCP Server Enhancements`

THINNER COVERAGE BELOW

06 **Granular feature access by resource group** NEW 55

Feature access controls can now be scoped at the resource group level rather than only by organization role, so permissions like Jobs, Inference Endpoints, and blog publishing can be granted independently per group.

— Explains scope and examples but no config path given `Granular Feature Access`

07 **Egress usage metrics for users and orgs** NEW 45

Users can now view their CDN egress usage directly in the Hugging Face dashboard, and organization dashboards additionally include a per-user egress breakdown showing how much data each member consumes.

— Describes what's shown but not exact dashboard path

`Egress metrics for users and organizatio…`

WAS THIS USEFUL? 👍 👎

Ollama

10 RELEASES · 2026-07-23 → 2026-08-15 NOTES ↗ CODE ↗

Get up and running with Kimi-K2.6, GLM-5.2, MiniMax, DeepSeek, gpt-oss, Qwen, Gemma and other models.

> Ollama's biggest window changes were new model support (Qwen3.8, Nemotron 3.5, Muse Glimmer, Laguna 2.1) paired with deeper agent-framework wiring via `ollama launch`, OpenAI-compatible `/v1/chat/completions` streaming fixes, and expanded GPU platform coverage including Windows ARM64 CUDA and B200 GPUs.

└──▷ WHAT SHIPPED · 13 FEATURES most completely described first ? what's the number?

01 **Nemotron 3.5 model and architecture support** NEW 88

Adds support for the Nemotron 3.5 prompt layout, selecting the 3.5 parser and renderer from its checkpoint template. Adds `nemotron-3.5-lightning`, a 30B mixture-of-experts model with 3B active parameters, runnable via `ollama run nemotron-3.5-lightning`, backed by a new Nemotron 3 architecture backend with MLX support for Nemotron 3 Nano Omni including Mamba2/recurrent layers, MoE routing, and quantized NVFP4/MXFP8 expert paths. Serves the model's multi-token prediction head as a built-in self-draft speculator, enabling speculative decoding without a separate draft model.

Run the new 30B Nemotron 3.5 Lightning MoE model locally for always-on agent workloads.

```
$ ollama run nemotron-3.5-lightning
```

— Detailed architecture and mechanism plus a runnable command v0.32.11 v0.32.9

02 **Qwen3.8 model family and chat template support** NEW 86

Adds Qwen3.8 model support via a dedicated renderer handling `reasoning-effort` and preserved-thinking chat template semantics, and folds leading system/developer prefixes into a single system turn so OpenAI-compatible coding agents that send `developer` role messages work unmodified. Adds `qwen3.8:27b` to the model library and a `qwen3.8:27b-mlx` Apple Silicon MLX-optimized tag for coding, professional work, research, and long-horizon agentic tasks. The Qwen renderer also now passes non-leading system messages through the raw ChatML path instead of returning an HTTP 500, supporting coding clients that inject runtime system prompts mid-conversation.

Run Qwen3.8 27B on Apple Silicon with the MLX-optimized variant for better throughput in coding-agent workflows.

```
$ ollama run qwen3.8:27b-mlx
```

Pull and chat with Qwen3.8 27B on any platform for long-horizon agentic or research tasks.

```
$ ollama run qwen3.8:27b
```

— Names model tags, template mechanism and runnable commands `v0.32.14` `v0.32.13` `v0.32.12`

---

03 **Muse Glimmer model support** NEW 84

Adds `muse-glimmer`, runnable via `ollama run muse-glimmer`, supporting coding agent integrations (Claude Code, Codex, Pi) and personal assistant frameworks (OpenClaw, Hermes) on all platforms including NVIDIA and AMD GPUs. Adds `muse-glimmer:30b-mlx`, a 30B multimodal variant runnable via `ollama run muse-glimmer:30b-mlx` on Apple Silicon through the MLX engine, with DFlash block-diffusion draft model support added to the MLX runner for speculative decoding and image input support added to the MLX runner for multimodal prompts (also enabling image input for Qwen3.5).

Run Muse Glimmer locally on Apple Silicon for direct chat or scripting via the REST API.

```
$ ollama run muse-glimmer:30b-mlx
```

— Names model tags and mechanisms; one run command given `v0.32.8` `v0.32.7`

---

04 **Agent framework launch integrations via ollama launch** NEW 80

Adds `ollama launch dsh` to launch DeepSeek Harness, DeepSeek's open-source agent harness, and `ollama launch muse` to launch Muse Code, Meta's agentic coding CLI. Adds `ollama launch claude --model muse-glimmer`, `ollama launch openclaw --model muse-glimmer`, and `ollama launch hermes --model muse-glimmer` to power Claude Code, OpenClaw, and Hermes with Muse Glimmer, then extends the same integrations to the Apple Silicon variant with `ollama launch claude --model muse-glimmer:30b-mlx`, `ollama launch pi --model muse-glimmer:30b-mlx`, `ollama launch openclaw --model muse-glimmer:30b-mlx`, and `ollama launch hermes --model muse-glimmer:30b-mlx`.

Launch DeepSeek Harness as a local agentic coding environment backed by Ollama.

```
$ ollama launch dsh
```

Launch Meta's Muse Code agentic coding CLI through Ollama.

```
$ ollama launch muse
```

Run Muse Glimmer locally as the backend for Claude Code for a fully local coding agent.

```
$ ollama launch claude --model muse-glimmer
```

Power the OpenClaw personal assistant framework locally with Muse Glimmer across WhatsApp, Telegram, Slack, and Discord.

```
$ ollama launch openclaw --model muse-glimmer
```

Launch Claude Code backed by Muse Glimmer on Apple Silicon for a fully local coding agent.

```
$ ollama launch claude --model muse-glimmer:30b-mlx
```

Launch the OpenClaw personal assistant framework with Muse Glimmer for a local AI assistant across WhatsApp, Telegram, and more.

```
$ ollama launch openclaw --model muse-glimmer:30b-mlx
```

— Nine exact launch commands across releases, all runnable `v0.32.11` `v0.32.8` `v0.32.7`

05 **OpenAI-compatible streaming compliance fixes** IMPROVED

78

Adds `stream_options.include_usage` support to `/v1/chat/completions` streaming, matching OpenAI's wire format: `role` only on the first chunk, `finish_reason` on its own trailing chunk, and usage in a separate chunk. Truncated responses now correctly report `finish_reason: "length"` instead of incorrectly reporting `"tool_calls"`.

Consume a streaming OpenAI-compatible chat response with token-usage stats, now that `stream_options.include_usage` is supported.

```
$ curl http://localhost:11434/v1/chat/completions \
  -H 'Content-Type: application/json' \
  -d '{
    "model": "qwen3.5",
    "stream": true,
    "stream_options": {"include_usage": true},
    "messages": [{"role": "user", "content": "Summarize the
OWASP Top 10"}]
  }'
```

— Exact fields named with a runnable curl example v0.32.6

06 **Agent TUI thinking trace and skill controls** NEW 76

Adds a `ctrl+o` keyboard shortcut to the agent TUI to toggle completed thinking trace details inline, and streams live thinking deltas beneath a `Thinking ↓ N tokens` row that collapses to a persistent `Thought` row once a response or tool call begins. Adds a permission/approval flow for model-initiated `skill` tool invocations while preserving direct user slash-skill activation without a prompt, and adds `/system` prompt inspection and on/off toggle commands with a cache-impact warning and completions while typing.

— Names exact shortcuts and commands but no full walkthrough v0.32.7 v0.32.4

07 **Expanded GPU platform support** IMPROVED 64

Adds CUDA support on Windows ARM64, enabling GPU-accelerated inference on ARM-based Windows devices. Adds B200 GPU support via CUDA 12 (compute capability 10.0 on Linux). Reduces memory use on Linux CUDA and ROCm iGPUs through Direct I/O (`dio`) enablement, and updates the MLX and llama.cpp engines.

— Names specific platforms, GPU model and mechanism v0.32.3

THINNER COVERAGE BELOW

08 **Namespace-qualified tool declarations in Responses API** IMPROVED 56

Expands namespace tool declarations in the OpenAI Responses API, unfolding nested `tools` arrays into namespace-qualified flat function names so namespaced tool calls are fully declared to the model.

— Names the API surface but no example call shown v0.32.7

09 **Automatic speculative decoding for Qwen3.5 on Apple GPUs**

IMPROVED 50

Qwen3.5 inference on Apple GPUs is faster: the MLX engine now automatically uses the model's MTP head for speculative decoding, without requiring a separate draft model or manual configuration.

— Mechanism named but no command to try it `v0.32.6`

10 **WebP image transcoding for vision requests** NEW 48

Automatically transcodes WebP image payloads to PNG before forwarding to `llama-server`, enabling vision model requests with WebP inputs, which were previously unsupported.

— Clear mechanism but no runnable example given `v0.32.14`

11 **Speculative decoding draft quantization alignment** IMPROVED 44

Quantizes draft-model output heads at the requested quantization type when creating speculative-decoding drafts, aligning draft and base model precision.

— Mechanism stated but no config or command given `v0.32.4`

12 **Laguna 2.1 model support** NEW 40

Adds chat, thinking, and tool calling support for Laguna 2.1 models, including a Metal inference fix.

— Names capabilities but no model tag or command `v0.32.3`

13 **Experimental image generation removed** BREAKING 37

Experimental image generation support has been removed; users who depend on it must remain on v0.32.5.

— Clear migration note despite thin description `v0.32.6`

└──▷ **BREAKING ON UPGRADE**

! Experimental image generation support is removed in v0.32.6. Workloads depending on it must stay on v0.32.5.

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

vMLX

9 RELEASES · 2026-07-20 → 2026-08-15 NOTES ↗

vMLX - JANGTQ Uber Compressed MLX Models - L2 Disk Cache (survives restart) + L1 Paged (super fast ttft) + Hybrid SSM Scheduler + Cont Batching + etc!

> vMLX shipped hybrid prefix caching that speeds up long-document follow-up turns by up to 20x and keeps 100k-token multiturn sessions stable, alongside a native DeepSeek V4 Flash runtime with a 29-58% prefill speedup, a process-wide SSD cache budget, and new model support for Qwen3.6-27B (automatic multi-token prediction) and the multimodal Muse Glimmer 30B.

└──▷ WHAT SHIPPED · 22 FEATURES most completely described first ? what's the number?

01 DeepSeek V4 Flash prefill speedup and retry resilience IMPROVED

90

A new indexed-attention prefill kernel (on by default) lifts DeepSeek V4 Flash prompt processing from 347 to ~449 tokens/s at 15k context (+29%) and from 199 to ~316 tokens/s at 40k context (+58%), sustaining 33+ tokens/s decode at both lengths. The engine now rewinds the live KV cache to the shared prefix on abort/retry instead of re-prefilling from scratch, dropping retry time-to-first-token at 15k context from 46s to 2.3s. The two-pass answer path reuses the first pass's KV cache, eliminating full prompt re-prefill on the second pass; KV pool quantization keeps cache RAM under 8GB even near the model's maximum context window. The engine also advertises the true hardware memory ceiling at startup, shrinks prefill chunks adaptively under Metal memory pressure, and returns a clean 413 for over-ceiling requests instead of crashing with a Metal OOM.

— Rich before/after numbers across multiple mechanisms v1.6.25

02 DSV4_ANSWER_RESERVE env var for answer budget control NEW

90

Adds the `DSV4_ANSWER_RESERVE` environment variable to control how many tokens are reserved for the answer portion of DeepSeek V4 Flash output — set to `0` to disable the budget split entirely. DeepSeek V4 Flash now automatically reserves part of the output budget for the answer when thinking mode is active, using a bounded reserve of up to 2048 tokens rather than a percentage, so long reasoning is not penalized at large token budgets.

Disable the answer budget split entirely if you want DeepSeek V4 Flash to use the full token budget for reasoning.

```
$ DSV4_ANSWER_RESERVE=0 vmlx serve <deepseek-v4-flash-model>
```

— Named env var with a runnable command example v1.6.23

03

Reasoning-effort controls across Anthropic, Ollama, and bundle defaults

IMPROVED

88

Ollama `think` string levels (`'low'` | `'medium'` | `'high'`) now enable thinking and select effort depth — previously these values were silently ignored. Anthropic `thinking.budget_tokens` now caps the reasoning chain at runtime — previously it only set a template hint with no runtime effect. Separately, Auto reasoning now aligns with each model bundle's native policy unless a request supplies an explicit supported thinking budget, while keeping reasoning-marker aliases out of visible content.

Cap reasoning token spend when calling vMLX via the Anthropic SDK — useful for cost/latency control on long agentic tasks.

python

```
import anthropic

client =
anthropic.Anthropic(base_url="http://localhost:8000/v1",
api_key="not-needed")
message = client.messages.create(
    model="local",
    max_tokens=4096,
    thinking={"type": "enabled", "budget_tokens": 2000},
    messages=[{"role": "user", "content": "Prove that sqrt(2) is
irrational."}],
)
print(message.content[0].text)
```

Set reasoning effort from an Ollama-compatible client to get measurably different think depth without changing the model.

```
$ curl http://localhost:8000/api/chat \
  -H 'Content-Type: application/json' \
  -d '{
    "model": "local",
    "think": "high",
    "messages": [{"role": "user", "content": "Explain the
halting problem."}]
  }'
```

— Named API fields with runnable examples and before/after v1.6.27 v1.6.18

04 **Qwen3.6-27B bundle support with automatic multi-token prediction**

NEW 85

Supports the Qwen3.6-27B bundle line (and the upcoming Qwen 3.8 line): the `qwen3_coder` tool-parser name now resolves to the XML-function parser, capability-only stamps route through the stock loader instead of the JANG codec, and the bundled multi-token prediction head is automatically constructed, quantized per per-module overrides, and engaged at the stamp's trained speculative depth — delivering 23.3 to 31 tokens/sec (+33%) on the 4-bit bundle with no flags required.

— Named parser and throughput numbers, works automatically v1.6.28

05 **Native DeepSeek V4 Flash runtime support** NEW 85

Adds native DeepSeek V4 Flash (DSV4) runtime support including DSML tool support, sampling guidance, cache-tier capabilities, and pool-quantization behavior derived from the selected bundle, plus explicit Activation-QAT control for DSV4 sessions with effective state reported in the app. Server and Chat settings now distinguish native DSV4 compiled decode and pooled-cache state from generic TurboQuant and unsupported whole-model cache modes. Cache handling preserves short append checkpoints, lossless L2 writes, valid eviction ancestry, partial-tail replay, SSD-only operation, RAM-to-SSD refault, and ratio-zero SWA rings and lossless q8 CSA/HCA pool segments through prompt snapshots and SSD reconstruction. Health output distinguishes the served model from the currently loaded model; explicit visible-final instructions after tool use enter a no-more-tools continuation pass with terminal abort cleanup drained before the runtime returns to idle; Electron settings warn when a DSV4 `top-p` value differs from bundle guidance and preserve remote model tool contracts. Release packaging attests the native Python DSV4 encoder, nested affine defaults, mixed module quantization metadata, and the clean JANG runtime source, with separate Sequoia and Tahoe packaging/notarization contracts bound to one source revision.

— Extensive named capabilities but no user command shown v1.6.20 v1.6.19

06 **Muse Glimmer 30B multimodal model support** NEW 80

Adds Muse Glimmer 30B support across all three JANG bundles (`JANG_2D` / `4M` / `6M`), including its text tower, windowed vision tower, recipient-routed reasoning rail, and ATEM tool dialect. Reasoning depth is now selectable in Chat Settings as Low / Medium / High / Extra High (previously only the Auto/On thinking toggle was exposed). The model correctly handles single- and multi-image prompts and reads video clips in temporal order, and each reasoning depth maintains its own prefix cache lineage so switching back to a previously used depth resumes that chain rather than starting cold.

— Names bundles and UI path but no runnable command `v1.6.27`

---

07 **Hybrid prefix caching and long-context stability** NEW 76

Hybrid prefix cache support accelerates follow-up turns on long documents by up to 20x — a 43.7k-token document follow-up drops from 107s to 5s, byte-identical output at temperature 0. Stable 100k-context multiturn is now proven: 97.6k-token conversations retrieve planted facts with 99.9% cache reuse and no memory faults. Chunked SSM re-derive, enabled by default, restores long-context cache reuse from 0% to 99.9% above 12.5k tokens on hybrid SSM model families.

— Concrete before/after benchmarks but automatic, no user action needed `v1.6.29`

---

08 **Process-wide SSD cache budget with LRU eviction** NEW 67

Adds a process-wide SSD cache budget with coordinated LRU eviction and off-request-path publication, plus detailed reporting of attempted prefixes, applied reuse, uncached suffixes, reconstruction, fallback prefill, and aggregate disk usage.

— Describes mechanism and reporting fields but no command `v1.6.18`

---

09 **Prefix cache index expanded to 262,144 tokens** IMPROVED 60

Prefix cache index expanded from ~64,000 tokens to 262,144 tokens; existing sessions are migrated automatically, so long conversations now get cache reuse instead of reporting a cold-start miss.

— Exact size figures and migration behavior given `v1.6.27`

---

10 **Per-model cache correctness and validation fixes** IMPROVED 60

MiniMax M2-family automatic cache storage now uses the correctness-first q8 policy, with bundle-owned calibrated TurboQuant settings and explicit user selections remaining authoritative. MiniMax M3 native sparse-cache blocks are now scoped to the prompt-prefill shape and its N-1 prompt-boundary contract, preventing a token-identical prefix produced under a different prefill shape from restoring numerically different

Lightning Indexer state. MiniMax tool arguments emitted as request-schema XML children are now accepted without exposing native tool markup as visible answer text. Compatible Nanbeige looped-transformer JANG bundles now validate their loop count, effective cache-slot count, and runtime metadata before generation, rejecting inconsistent bundles rather than running with a silently short cache, and their cache namespaces now include the effective repeated-layer layout to prevent reuse between incompatible loop counts. Laguna tool-result history, incomplete terminals, mixed-SWA cache reuse, and affine-JANG JIT policy are hardened.

— Covers three model families' fixes without user-facing controls `v1.6.19` `v1.6.18`

THINNER COVERAGE BELOW

11 **macOS Sequoia/Tahoe packaging and signing** IMPROVED 55

Separate Apple-silicon downloads are now provided for macOS Tahoe and for Sequoia-compatible systems. Packaging is strengthened so bundled Python, JANG dependencies, release metadata, and artifacts remain tied to one source revision. Both macOS Sequoia and Tahoe release artifacts are Developer ID signed, notarized, stapled, and Gatekeeper-accepted for out-of-the-box installation without security prompts.

— Names OS versions and signing guarantees across three releases `v1.6.19` `v1.6.18` `v1.6.14`

12 **Request parameter handling fixes in Ollama-compatible API** IMPROVED 54

Ollama requests now preserve an explicitly supplied `top_k`, including `0`, instead of silently dropping it during request translation. Automatic prompt limits are now derived from the configuration of the model that is actually loaded, bounded by the model's declared context ceiling and current memory budget.

— Names the parameter fixed but no example call `v1.6.19`

13 **Architecture-aware KV cache behavior** NEW 49

Adds architecture-aware KV cache behavior covering prefix, paged, block-disk, and TurboQuant cache policies, with model-family gates keeping cache settings explicit and model-safe.

— Names cache policy types but no configuration surface `v1.6.14`

14 **Tool-call continuation and enforcement** IMPROVED 48

Bounded Electron tool workflows now better enforce requested exactly-once tool calls while leaving other explicitly requested tools available until their results have been returned. Media is now retained across Responses tool continuations, with local attachment playback permitted in the Electron desktop app. Structured tool continuation additionally allows tool-call sequences to resume and chain within a single inference pass.

– Behavioral changes across releases without concrete controls v1.6.19 v1.6.18 v1.6.14

15 **Bundle-owned generation defaults for sessions** IMPROVED 45

New and restored Electron chats now hydrate temperature, Top P, Top K, repetition penalty, and output limits from the selected model bundle without presenting missing values as saved zero-valued overrides. Fresh sessions likewise derive sampling, parser, output, template, and native MTP settings from the selected model bundle without converting inherited values into hidden saved overrides.

– Names the settings fixed but no config path given v1.6.19 v1.6.18

16 **Expanded per-family model support** IMPROVED 42

Verified per-family cache, parser, and generation-default support now spans 12 model families, including DSV4 Flash native composite caching, Gemma 4 mixed-SWA with audio, Qwen 3.6/3.8 hybrid lines with native MTP, and TurboQuant KV bundles.

– Names families but not per-family mechanism v1.6.29

17 **Paged RAM cache parity for vision models** IMPROVED 40

Vision models no longer use a slower cache path in the app — paged RAM is now enabled consistently between the app and the command line, restoring KV reuse for vision sessions.

– States the fix but no metrics or mechanism detail v1.6.27

18 **JANG 2.5.34 dependency requirement** BREAKING 40

This release requires JANG 2.5.34, a hard dependency bump for anyone running vMLX v1.6.18 or later.

– Clear version requirement but no migration steps given v1.6.18

19 **Engine internals: native-MTP telemetry and cache encoding path** IMPROVED 35

Native-MTP health and profiling now expose bounded cache-lifecycle, acceptance, and phase-timing telemetry without changing generation policy. Persistent cache payload encoding is moved off the inference-completion path after tensors are safely detached from MLX, reducing synchronous CPU work at the end of a request.

– Internal engine changes described only in prose v1.6.19

20 **Localization and rendering stability** IMPROVED 30

New settings and compatibility messages are localized in shipped English, Spanish, Japanese, Korean, and Chinese interfaces. KaTeX, Markdown, HTML, XML, currency, code-fence, and in-progress reasoning rendering are improved, and localized UI context is stabilized during development reloads.

— Lists languages and surfaces but no mechanism or metric `v1.6.19` `v1.6.18`

21 **Multimodal conversation-state isolation** NEW 25

Adds multimodal conversation-state isolation, enabling independent context tracking across modalities in a single session.

— Described only at a high level, no mechanism `v1.6.14`

22 **Progressive reasoning and content streaming** NEW 25

Adds progressive reasoning and content streaming, delivering incremental output during multi-step inference.

— Bare description with no mechanism or metric `v1.6.14`

╰──▷ BREAKING ON UPGRADE

! This release requires JANG 2.5.34.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · August 19, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe