THE AI TOOLCHAIN

NO. 052

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED OCTOBER 9, 2026 · EVERY DAY

The daily firehose — everything the toolchain shipped today, already filtered.

▸ // HOW THIS ISSUE IS MADE SHOW

VIEW

ISSUE VIEW

scroll



SCROLL | EXPAND IN PLACE | PREVIEW PANE

INFORMATION DENSITY

everything



BRIEFED | EVERYTHING

Every tool in the issue, in full.

all 219 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 40 tools

▾ // THE BRIEF HIDE

What stands out across today's releases, grouped by what it lets you do. Every tool named links to its entry below.

Three unrelated things stand out. TRL's fused LM head is on by default and drops peak VRAM 52–82% at 8k tokens, so long-context fine-tunes hardware that refused them yesterday. Hayhooks 2.0 checkpoints

vim ? help

pipelines in Redis so a run survives a restart. And LiteLLM's OTel v2 routes traces per tenant into SigNoz.

BUILD

Fine-tune on longer sequences without renting a bigger GPU

TRL v1.15.0 turns on a fused LM head by default across SFT, DPO, KTO, GRPO and RLOO — the logits tensor over a full vocabulary is the single biggest activation in a training step, and fusing it avoids materializing it in full. Peak VRAM falls 52–82% at 8k tokens and sequences up to 6.9× longer fit, so jobs that OOM'd now run unchanged. Hugging Face also shipped OpenEnv for isolated agentic RL environments.

TRL · Hugging Face

BUILD

Resume a long-running pipeline after a crash instead of restarting it

Hayhooks 2.0 checkpoints pipeline state to Redis and makes SSE streams reattachable, so a human approval step or a dropped client connection no longer forces the whole run to start over. Multi-step agent work that spans minutes becomes something you can deploy behind a load balancer rather than a single process you must not restart.

hayhooks

OPERATE

Send each customer's traces to their own backend without a second collector

LiteLLM's OpenTelemetry v2 adds native SigNoz export and per-tenant trace routing, so a multi-tenant gateway can keep one customer's prompt traffic out of another's observability stack; the `litellm-core` distribution also trims mandatory dependencies for a 32.2% smaller install. Braintrust's `span_customizers` hook redacts and tags spans before they leave the process, which is the step teams otherwise hand-roll to keep PII out of a trace store.

LiteLLM · Braintrust

DEPLOY

Serve long-context attention cheaper on CUDA, and get the same answer sooner

ONNX Runtime's CUDA Plugin EP adds SparsePagedAttention, DynamicSparseAttention and INT4 paged KV caches — paging plus 4-bit keys and values is what lets a long-context batch stay resident instead of

> thrashing memory. SGLang's speculative decoding gains block-level verification and a LiLiCorr drafter, so more drafted tokens clear verification per pass and latency drops on the same hardware.
>
> ONNX Runtime

BUILD

◆ AI Coding Agents

Warp

Feedback ⓘ 2026.10.07 (v0.2026.10.07.08.29) NOTES ↗ RANK

Warp is an AI terminal that runs commands and helps developers build software.

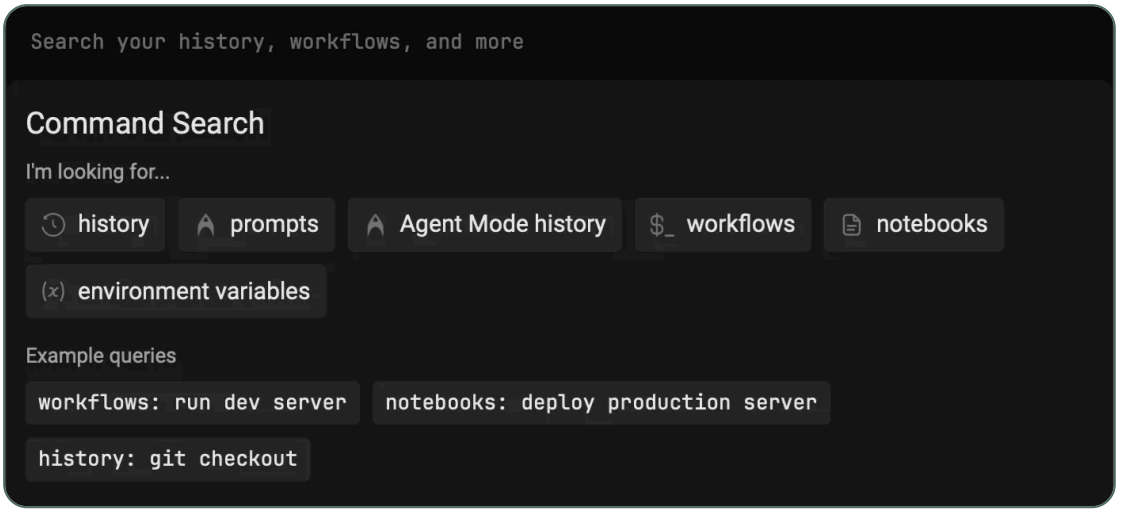
> Warp's biggest shipment this window is a new Automation Platform — an HTTP API plus Python/TypeScript SDKs for persistent, triggered multi-agent workflows — alongside a unified Command Search panel, auto-fixing Command Corrections, enterprise BYOLLM key sharing and an Analytics API, and several CLI/runner refinements.

└──▷ WHAT SHIPPED · 10 FEATURES most completely described first ? what's the number?

01 Unified Command Search with category filters IMPROVED

85

Command Search now offers a single panel across shell history, Workflows, Prompts, and agent conversations, with filters `history:` / `h:` / `H-TAB`, `prompts:` / `p:` / `P-TAB`, and `ai_history:` / `a:` / `A-TAB` to scope results, results ranked by relevance, and a horizontally resizable panel. When fzf is installed, Warp hands `Ctrl+T` (file search) and `⌥+C` / `Alt+C` (directory search) to the fzf widget, leaving Command Search reachable via the Command Palette.



Scope Command Search to agent conversation history only — useful when you want to recall a specific AI-assisted session without wading through shell history.

📍 1. Press Ctrl+R to open Command Search. 2. Type 'a:' followed by your query (e.g. `a: kubernetes deploy`) to filter results to Agent Mode history only. 3. Press Enter to insert the selected command into the Input Editor.

Recover Warp's unified Command Search panel when fzf is installed and has rebound Ctrl+R.

📍 1. Open the Command Palette (Cmd+P on macOS / Ctrl+P on Windows/Linux).
2. Search for `Command Search` and select it to open the unified panel (history, Workflows, Prompts, agent conversations).

— Names exact filter syntax and hotkeys with step-by-step usage. product docs

---

02 **Command Corrections auto-fix suggestions** NEW 85

After a failed command, Warp surfaces an auto-correct suggestion panel above the Input Editor — press `→` or click to apply the fix. Correction rules cover `brew`, `cargo`, `cat`, `cd`, `chmod`, `conda`, `cp`, `docker`, `git`, `go`, `grep`, `java`, `ls`, `mkdir`, `npm`, `pip`, `python`, `sed`, `sudo`, `yarn`, plus a generic misspelling rule, and can be toggled via Settings > Features > Terminal Input > `Suggest corrected commands` (enabled by default).

Disable Command Corrections if auto-suggest panels interrupt your workflow.

📍 In Warp, go to Settings › Features › Terminal Input and toggle off `Suggest corrected commands`.

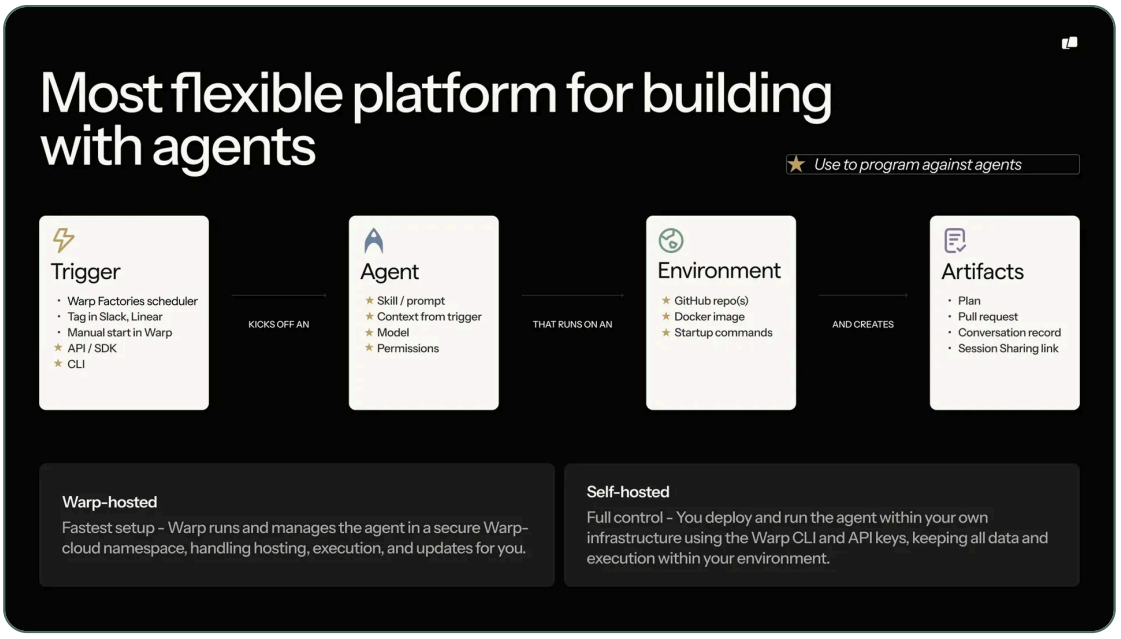
— Lists every covered command and the exact settings path. product docs

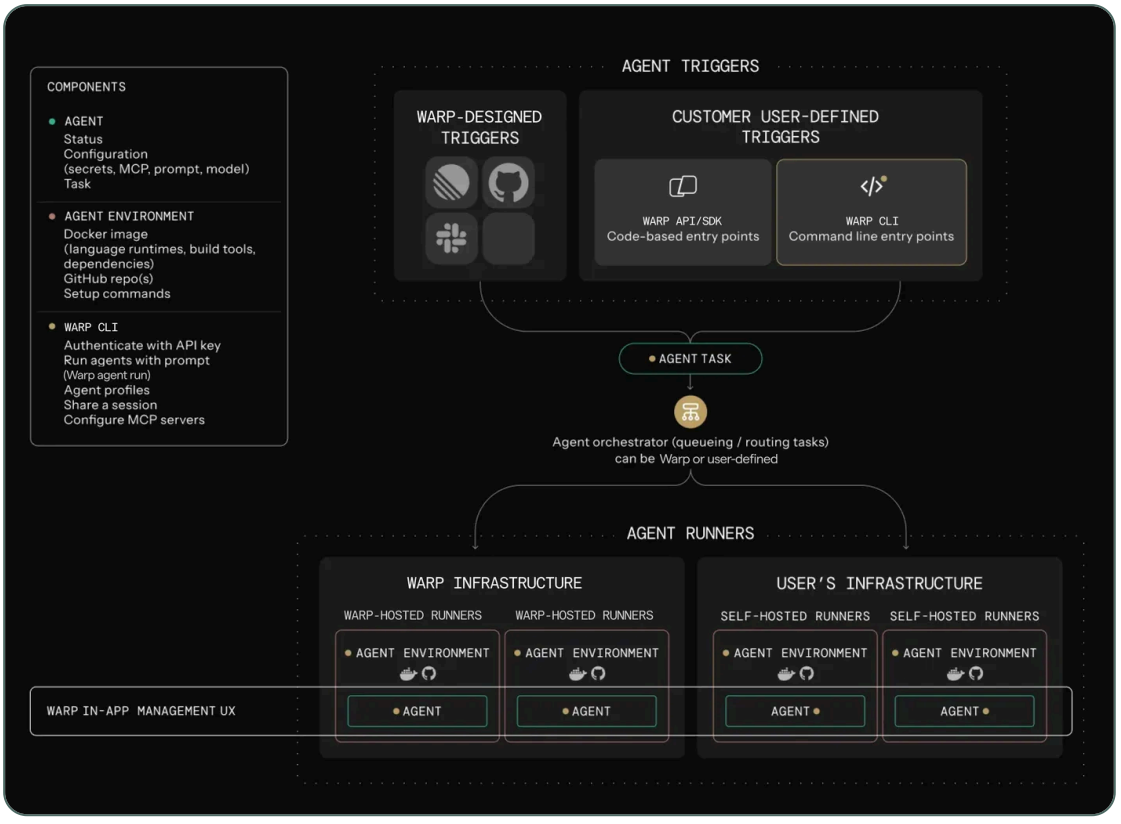
---

03 **Automation Platform: HTTP API, SDKs and triggered cloud agents**

NEW 80

The Warp Platform API lets teams create and inspect tasks over HTTP — submit a prompt with optional configuration, poll status, and fetch results with full provenance — backed by the `warp-platform-sdk` Python package and `@warp-dot-dev/warp-platform-sdk` TypeScript package, both offering typed requests/responses, built-in retries, and consistent errors. The resulting Automation Platform runs agents against defined tasks and triggers — schedules, integrations, or APIs — to handle background work such as triaging new issues as filed, starting a fix the moment CI fails, or pull request review, enabling persistent multi-agent workflows where specialized cloud agents move work items through triage, specification, implementation, and review.





-
-
-

— Names API shape and both SDK packages but no runnable example. `product docs`

---

04 **Default runner flag for environments** NEW 80

The `oz environment update --default-runner <RUNNER_UID>` command sets an environment's default runner from the CLI, so subsequent cloud agent runs use it automatically without manual selection.

Pin a specific runner as the default for an environment so all subsequent cloud runs use it without manual selection.

```
$ oz environment update --default-runner <RUNNER_UID>
```

Pin a specific runner as the default for an environment so all agent runs start there automatically.


```
$ oz environment update --default-runner <RUNNER_UID>
```


— Exact runnable command with its flag and argument. `2026.10.07 (v0.2026.10.07.08.29)`

THINNER COVERAGE BELOW

05 **Shared LLM keys and custom endpoints (BYOLLM)** NEW — 45

Business and Enterprise teams can share provider API keys and custom LLM endpoints across the team for both interactive sessions and cloud agents via Bring Your Own LLM (BYOLLM).

— Names the BYOLLM mechanism but gives no setup steps. `product docs`

06 **Cloud agent run diagnostics improvements** IMPROVED — 40

Codex run usage now reports cache-write token counts when available, and cloud runs report which specific environment setup command timed out instead of a generic sandbox-expiry failure.

— Describes two diagnostic improvements without a surface to act on. `product docs`

07 **Kiro CLI in coding-agent toolbar** NEW — 30

Warp adds Kiro CLI support to its third-party coding-agent toolbar and rich input.

— Names the integration but no usage detail. `2026.10.07 (v0.2026.10.07.08.29)`

08 **Analytics API for Business/Enterprise** NEW — 25

Warp introduces an Analytics API available to Business and Enterprise plans.

— Named but undescribed — no endpoint, method, or data shape given. `product docs`

09 **Standalone Warp Agent CLI** NEW — 25

The Warp Agent can now run in any terminal via a standalone CLI, rather than only inside the Warp desktop app.

— No CLI name, install method, or usage given. `product docs`

10 **Video recording for Windows computer-use agents** NEW — 25

Computer-use agents running on Windows cloud runners can now record video of their sessions.

— States the capability only, no access path given. 2026.10.07 (v0.2026.10.07.08.29)

WAS THIS USEFUL? 👍 👎

Augment Code

Feedback ⓘ snapshot-20261009 NOTES ↗ RANK

Augment Code is an AI coding assistant for VSCode and JetBrains IDEs that provides code review and context-aware suggestions, with Cosmos as its enterprise platform.

> Augment Code's Cosmos platform gains organization-wide model governance and secrets management, user groups for access control, agent-initiated private Slack channels, and several Session and connector reliability improvements.

└──▷ **WHAT SHIPPED · 10 FEATURES** most completely described first ? what's the number?

01 Model access controls and fleet compliance flagging NEW

70

Adds a Model Access page where admins can set a default model and override access per individual model for their organization, paired with Fleet Management flags that surface Experts still using a model the organization no longer allows.

Restrict your organization to approved models and catch any Expert still using a disallowed one.

> 📍 In Cosmos, go to Settings › Model Access, set your organization default model, then toggle off any models you want to block. Open Fleet Management to review Experts flagged for using a disallowed model.

— Names both pages with clear steps but no deeper mechanism given. snapshot-20261009

02 Agent-created private Slack channels and group DMs NEW

70

Enables agents to create private Slack channels and start group DMs, which requires a Slack workspace admin to reconnect via `Settings > Connectors` to grant the new permissions.

Allow agents to open private Slack channels and group DMs for incident response workflows.

> 📍 In Cosmos, go to Settings › Connectors, locate your Slack workspace, and have a Slack workspace admin reconnect Slack to grant the new channel and DM permissions.

— Names exact reconnect path and requirement, a clear setup action. snapshot-20261009

THINNER COVERAGE BELOW

03 Organization secrets management in Secrets IMPROVED

55

Enables admins to manage every organization secret — including owner, assigned Spaces, and deletion — from Secrets.

— Names the fields and surface but no exact navigation steps given. snapshot-20261009

04 **Browser push notifications for agent updates** NEW 55

Adds browser push notifications for agent updates, available in beta via Settings; enable it from the Notifications section to get real-time alerts when an agent completes a task or needs input without staying on the Cosmos tab.

Get real-time browser alerts when an agent completes a task or needs input, without staying on the Cosmos tab.

> 📍 In Cosmos, open Settings, find the Notifications section, and enable Browser push notifications for agent updates (currently in beta).

— Exact settings path and beta status given, mechanism is simple. snapshot-20261009

05 **MCP catalog and connector management improvements** IMPROVED 55

Adds Runlayer to the MCP catalog, adds filtering for connectors that need attention, and surfaces MCP errors that include remediation guidance.

— Names all three additions but no steps to use them. snapshot-20261009

06 **Incident Investigator accepts typed descriptions** IMPROVED 50

Extends Incident Investigator to accept incident descriptions typed directly in Cosmos, not only from alert threads.

— Clear before/after but no concrete steps given. snapshot-20261009

07 **Session persistence improvements** IMPROVED 45

Keeps Sessions active while a tunnel is in use and extends Session snapshot retention to 14 days.

— Gives a concrete retention number but no further mechanism. snapshot-20261009

08 **Multi-org GitHub cloning in a Session** NEW 40

Supports cloning repositories from more than one GitHub organization within a single Session.

— States the capability with no further mechanism or steps. snapshot-20261009

09 **Environments self-repair repositories** IMPROVED 40

Enables Environments to repair their own repositories, with failed refreshes now surfacing actionable next steps.

— Describes behavior change but no UI path or command. `snapshot-20261009`

10 **User groups for access management** NEW 35

Enables admins to create user groups and assign them to resources for access management.

— Bare description of the capability with no mechanism or UI path. `snapshot-20261009`

└──▷ **ALSO FROM THESE RELEASES**

WAS THIS USEFUL? 👍 👎

Amazon Kiro Crew

Feedback ⓘ v0.8.0 NOTES ↗ RANK

Kiro Crew is an open-source workspace for coordinating AI coding agents.

> Kiro Crew's v0.8.0 release adds granular memory and session-title controls, bundles its own agent CLI into desktop installers, and strengthens credential and link safety with new redaction and blocked-link cards, alongside a smarter monitoring-loop wake judge, reorganized sidebar and settings, and preview features for team grouping and Jev-based routing.

└──▷ **WHAT SHIPPED · 28 FEATURES** most completely described first ? what's the number?

01 **Memory distillation controls in config.json** NEW 85

Adds `memory.persistence_enabled` in `config.json` to stop the gateway distilling sessions into memory entirely, and `memory.inject_memory`, `memory.inject_lessons`, and `memory.inject_activity` in `config.json` to independently drop the memory, lessons, and activity blocks from a turn.

Prevent the gateway from writing any memory after sessions, and strip activity context from every turn, to keep sensitive conversations ephemeral.

json

```
{
  "memory": {
    "persistence_enabled": false,
    "inject_activity": false
  }
}
```

— Exact config keys shown with a working JSON example v0.8.0

02 **Auto-refreshing session titles** NEW 75

Adds `dashboard.title_refresh_every_turns` in `config.json` to re-examine and rename auto-generated session titles every N turns when the topic has drifted.

Keep session titles accurate in long-running investigative chats by re-evaluating the title every 10 turns automatically.

```json
{
  "dashboard": {
    "title_refresh_every_turns": 10
  }
}
```

— Exact config key with example value supplied v0.8.0

03 **Crew log recorded by default** BREAKING 75

Crew log is now recorded by default for every chat without any configuration, enabling session ledgers, conductor work boards, and Issue Radar on a fresh install; set `KIROCREW_CREW_LOG=0` to disable the whole crew log if session data should not be written.

— Names env var and the default-behavior change it reverses v0.8.0

04 **Blocked-link card with host allowlist** NEW 75

Adds a blocked-link card naming the destination and rule, with Open once, Allow for this host, and Inspect actions, and lists allowed hosts under Settings → Security → Allowed link hosts.

— Exact settings path and all card actions named v0.8.0

05 **Team grouping for crewmates (preview)** NEW 75

Adds team grouping for crewmates (preview) under Settings → Developer → Feature Previews, with a New team option showing per-crewmate activity, pending questions, and weekly activity.

— Exact settings path and preview contents named v0.8.0

06 **Credential redaction toggle for file views** NEW 70

Adds a Credential redaction in file views toggle under Settings → Security (on by default) that unmasks files the dashboard opens while keeping chat, channels, and history redacted.

— Exact toggle location and default state given v0.8.0

07 **Sidebar organization: folder order, conductor view, tag permissions**

NEW 70

Adds Folder order choice to the sidebar sort menu (custom arrangement, by name, or by date created); a conductor view in the sidebar that nests each session under the one that opened it, showing child and needs-you counts on collapsed rows; and a per-tag

Agent permissions choice (Human only, Agent: add only, Agent: add and remove) in the sidebar's Manage Tags panel.

— All three sidebar additions and their options named v0.8.0

08 **Native file delivery to WeCom** IMPROVED 70

A file sent to WeCom now arrives as its native type: PNG/JPEG as pictures (capped at 2 MB), MP4/WebM as videos (capped at 20 MB), and other files as downloads (capped at 20 MB).

— Exact file types and size caps named v0.8.0

09 **Smarter monitoring loop and PR watch wake-ups** IMPROVED 65

Adds a wake judge that screens monitoring-loop ticks and wakes you only when a decision is needed, costing no model turn when nothing is new; a monitoring loop watch now ends automatically on a merged or closed pull request; and a pull request watch now wakes the agent on a review or comment requesting a change or asking a question, not only on a failed check.

— Mechanism described in detail, no exact command v0.8.0

10 **Message Font Size stepper** NEW 65

Adds a Message Font Size stepper under Settings → Chat → Transcript to scale text, code, diffs, tables, chips, and the composer across all sessions.

— Exact settings path named for the new control v0.8.0

11 **Agent templates tab in Agent Capabilities** NEW 65

Adds an Agent templates tab under Agent Capabilities to view, create, duplicate, edit, or delete templates grouped by origin, and start a one-off chat or enroll as a crewmate.

— Exact UI location and full set of actions named v0.8.0

12 **Stub-command review for shared MCP backends** NEW 65

Adds a stub-command review under Developer → MCP Management that shows the exact command and environment a shared backend would launch before any change takes effect.

— Exact UI path and safety mechanism named v0.8.0

13 **Settings UI reorganization and search** IMPROVED 65

Settings → Chat is reorganized into a named-group rail: Transcript, Composer, Sessions, Side panel, Model, About You, Discovery, and Advanced; a pinned search box is added at the top of the Settings sidebar on desktop for finding settings while tabs scroll.

— All setting groups and the search feature named v0.8.0

14 **Lock tag reveals redacted credential source** NEW 60

Adds a lock tag in redacted replies that opens a card naming the matched credential, its source file or command, and how to reach the value safely.

— Mechanism named, no exact navigation path v0.8.0

15 **Agent-driven session management: model, pin, fork** NEW 60

Enables an agent to set a session's model at open time or move an idle session to another model before its next turn, to pin or unpin a live session at the top of the sidebar following the same ownership rules as filing or tagging, and to fork sessions so a new session inherits the dashboard Fork button's state while the agent can nest or release sessions in the sidebar.

— Three agent capabilities named without exact commands v0.8.0

THINNER COVERAGE BELOW

16 **Quiet check-ins for monitoring-loop judge (preview)** NEW 55

Adds a Quiet check-ins row on the Decisions (Jev) card (preview) so a monitoring loop's judge runs on the machine's existing model provider without a separate Jev key.

— Mechanism named, no exact setup steps given v0.8.0

17 **Add Remote Crew from launch failure dialog** NEW 55

Adds Add Remote Crew option in the launch failure dialog, opening a four-field form whose Save and Retry connects to the remote crew immediately.

— Names dialog, form size, and resulting action v0.8.0

18 **Daily History table in Usage drill-in** NEW 55

Adds a Daily History table in the Usage drill-in showing each day's credits used and its share of the plan allowance.

— Exact table and metrics it displays named v0.8.0

19 **Minimap relocation and scrubbing** IMPROVED 50

Adds a Minimap location setting to move the turn minimap to the right edge; the minimap now supports press-and-drag scrubbing and hover preview of turns.

— New interactions described, no exact setting path v0.8.0

20 **Folder-level steering directories** NEW 50

Adds folder-level additional steering directories so every chat filed under a folder or its subfolders starts with that steering on any agent backend.

— Clear scope but no exact config surface named v0.8.0

21 **Facts recalled and context breakdown chart** NEW 50

Adds a Facts recalled for this task row and a stacked area chart of context breakdown over the session, visible with Developer Mode on.

— Names UI element and its visibility condition v0.8.0

22 **Requests tab in Code Review Sage** NEW 50

Adds a Requests tab in Code Review Sage listing open pull requests across repositories that ask for your review.

— Names UI tab and its scope v0.8.0

23 **Bundled agent CLI in desktop installers** IMPROVED 45

Bundles its own agent CLI into the macOS, Linux, and Windows desktop installers so the agent runs after sign-in alone with no separate setup.

— Describes benefit without an exact install command v0.8.0

24 **Crewmate display name field** NEW 45

Adds a Display name field on the crew form under Agent Capabilities → Agents to label a crewmate across the dashboard independently of its id.

— Exact UI path given but a thin single-field addition v0.8.0

25 **Conductor support in Discord and Telegram DMs** NEW 45

Enables Discord and Telegram DMs to run a conductor that dispatches workers and uses the work ledger.

— Surfaces named but no setup detail given v0.8.0

26 **Sub-agent Approve/Deny buttons in Discord** NEW 45

Adds sub-agent Approve and Deny buttons directly in Discord channels, threads, and DMs.

— Thin one-line addition with named surfaces v0.8.0

27 **MCP app messages labelled in chat** NEW 40

Enables MCP app buttons and form submissions to send messages into the chat labelled Sent by the app.

— Behavior named but no setup detail given v0.8.0

28 **Auto (Jev) routing (preview)** NEW 35

Adds Auto (Jev) routing (preview) so any session left on Auto is routed per turn when the switch is on.

— Thin description of a preview toggle `v0.8.0`

└──▷ BREAKING ON UPGRADE

! The crew log is now recorded for every chat by default; set `KIROCREW_CREW_LOG=0` to disable it if you do not want session data written.

WAS THIS USEFUL? 👍 👎

yetone magpie

Feedback ⓘ v0.1.1134 CODE ↗ RANK

A macOS menu-bar app, TUI, and CLI that lists every AI coding agent on the machine and switches each one's model, such as Codex on DeepSeek or Claude Code on Kimi, by rewriting only that key in the agent's own config.

> Magpie's busiest window added per-profile agent model lists, a `SetOnlyPicked` lock for agent model selection, Gemini API support as a custom provider, and a `magpie quota` CLI command, alongside reliability fixes for Claude Code turn handling and routing-page UI improvements.

└──▷ **WHAT SHIPPED · 22 FEATURES** most completely described first ? what's the number?

01 **Lock agents to explicitly picked models** NEW 80

Adds a `SetOnlyPicked` control and `PickedModels` provider function so an agent's model list is frozen to only the models ticked for it — models added later, from existing or new providers, are excluded until explicitly picked.

— Names exact control and function, no UI path given v0.1.1130

02 **Gemini API support for custom providers and detection** NEW 80

magpie supports the Gemini API (`generateContent`) as a custom provider Base URL (e.g. `…/v1beta`), listing models from it and sending the API key in the `x-goog-api-key` header; provider auto-detection also probes this `…/v1beta` endpoint with the same header to select a Gemini model from the candidate list, and the editor shows a Detect option when a Gemini model is selected in the provider picker.

— Names exact endpoint and header for setup v0.1.1127 v0.1.1125

03 **Private-repo skill installs via GitHub token** NEW 80

Installs skills from private GitHub repositories using the GitHub token set in Settings → Network and sharing, or the `GITHUB_TOKEN` / `GH_TOKEN` environment variable.

Install a skill from a private GitHub repo by supplying a GitHub token so Magpie can authenticate against GitHub's API.

```
$ export GITHUB_TOKEN=ghp_yourTokenHere
```

— Exact settings path, env vars, and runnable example v0.1.1123

04 **DO_NOT_TRACK environment variable** NEW 75

Setting `DO_NOT_TRACK=1` before launching magpie disables all telemetry reporting, useful for unattended or air-gapped deployments.

Disable all telemetry in unattended or air-gapped deployments by setting the environment variable before launch.

```
$ DO_NOT_TRACK=1 magpie
```

— Exact env var and launch command given `product docs`

05 **Collapsible groups on Routing page** NEW 75

Provider accounts and nested groups on the Routing page can now collapse, with groups of 6 or more starting folded; the open/closed state persists per provider and model using `magpie.routingKeysOpen` and `magpie.routingFolded` in localStorage.

— Names exact localStorage keys and fold threshold `v0.1.1133`

06 **WebDAV sync conflict detection by file kind** IMPROVED 70

WebDAV sync now identifies non-backup server files by kind (`empty` , `cut` , `page` , `xml` , `format` , `json` , `other`) and offers to replace them with the local setup.

— Names exact file kinds but no step-by-step flow `v0.1.1134`

07 **Claude Code mid-turn disconnect recovery** NEW 70

If a client disconnects mid-turn, the gateway now recovers the Claude Code run by taking back only that turn rather than the whole conversation, then tells a fresh Claude Code session which turns were already answered and which turn to answer next.

— Clear mechanism but no command to invoke it `v0.1.1133`

08 **SSE keepalives for queued streamed requests** IMPROVED 70

Sends the stream's HTTP 200 response and SSE keepalives to the agent after 15 seconds when a streamed request is waiting for its key's concurrency slot or requests-per-minute allowance, preventing premature client timeouts.

— Names exact timing but no toggle or config `v0.1.1123`

09 **Context-window fill indicator in routing card** NEW

65

A live context-window fill bar appears in the folded routing card, one line high, showing content and fill level from the active request; it folds to its header with a fill indicator (e.g. `114K / 272K · 42%`) that persists across requests and page reloads.

— Gives example format but no adjustable setting v0.1.1127 v0.1.1126

10 **Per-partner daily engagement telemetry** NEW

65

Tracks per-partner daily engagement counts (`PartnerShown`, `PartnerOpened`, `PartnerKeys`, `PartnerAdded`), capped per day, and sends them with existing daily stats only for ended days.

— Names exact counters but no opt-out mechanism noted v0.1.1126

THINNER COVERAGE BELOW

11 **Usage heatmap on Requests tab** NEW

50

Adds a 53-week heatmap on the Requests tab, viewable by tokens, requests, or cost.

— Names metric views but little navigation detail v0.1.1134

12 **Per-profile agent model lists** NEW

50

Each agent's model list is now stored inside a profile, so switching profiles also switches each agent's model selection.

— Explains mechanism without naming a config key v0.1.1133

13 **Verbatim passthrough of thinking tokens and tool call IDs** IMPROVED

50

Signed thinking tokens and tool call IDs are now passed back to the vendor exactly as received, preventing 400 errors caused by redaction of those values.

— Explains the fix but no named endpoint or flag v0.1.1133

14 **Discover button for Skills and MCP marketplace** NEW

50

Adds a Discover button to the Skills and MCP tabs' strip, linking directly to the marketplace.

— Names exact button and location v0.1.1125

15 **Isolated runs for Claude Code fork sub-agents** IMPROVED

50

A request whose tool results were already answered by another request (e.g. Claude Code fork sub-agents) now gets its own Claude Code run instead of reusing the earlier

response.

– Explains scenario but no config switch v0.1.1123

16 **Token-speed suppression for bursty replies** IMPROVED 50

Suppresses token-speed display for replies delivered in a burst over a short window (e.g. a couple of hundred milliseconds), avoiding misleadingly high rates like 2,237 tok/s.

– Gives example rate but no adjustable threshold v0.1.1123

17 **Cross-platform distribution across macOS, Windows, Linux** NEW 45

magpie now runs as a menu bar app, window, TUI, web UI, and CLI on macOS, Windows, and Linux, as well as in Docker and Termux.

– Lists platforms and forms but no setup instructions product docs

18 **Round-robin spread for simultaneous conversations** IMPROVED 45

Under `In turn` routing, simultaneous new conversations are now spread across all accounts instead of funneling them to the single least-held account.

– Describes before/after but no config toggle v0.1.1126

19 **Codex app version shown on Agents page** IMPROVED 45

The Agents page now shows Codex's desktop app version alongside its CLI version (e.g. `CLI 0.130.0`, `App 26.930.31730`).

– Example values given but simple UI addition v0.1.1123

20 **New-partner notification on Add provider button** NEW 40

Shows a dot on the `+ Add provider` button and the partner name in the title when a new partner has been listed since the add sheet last appeared.

– Simple UI cue with minimal mechanism v0.1.1126

21 **MLX-Serve local-server provider preset** NEW 35

Adds MLX-Serve as a local-server provider preset alongside the existing oMLX preset.

– Bare addition with no setup detail v0.1.1134

22 **Agent version change display in updates** IMPROVED 35

The update summary now displays each agent's version change (from/to) after an agent update.

— Thin UI addition with minimal detail v0.1.1133

WAS THIS USEFUL? 👍 👎

claude-mem

Feedback ⓘ v13.35.0 NOTES ↗ RANK

Gives coding agents persistent memory by capturing session activity, compressing it with a model, and injecting relevant context into later sessions.

> claude-mem v13.35.0 adds a guided mem_search retrieval tool with strict size limits, a save_memory MCP tool for explicit writes, a native memory hook bridge that intercepts built-in tool reads to inject context, and an opt-in Markdown note watcher, alongside hosted-runtime routing, cleaner MCP replies, and installer/plugin improvements.

└──▷ **WHAT SHIPPED · 10 FEATURES** most completely described first ? what's the number?

01 mem_search progressive disclosure and auto mode NEW

95

The `mem_search` MCP tool gains a guided mode with a 3-step progression (`mem-search step 1 of 3` / `step 2 of 3` / `step 3 of 3`) plus `mode: "auto"` for single-call ranked lexical retrieval. `limit` is capped at 20, `maxDetails` defaults to 3 (max 5), `depthBefore` / `depthAfter` default to 2 (max 3), queries are capped at 500 characters/1 KiB UTF-8, and pagination cursors are 27-character opaque tokens (`ms_` + 24 URL-safe characters) with 15-minute server-side expiry.

Run a single-call automatic memory search for a topic without multi-step agent coordination — useful in CI agents or scripts where you want one bounded retrieval with no follow-up calls.

```
$ # Inside an MCP-connected agent, invoke mem_search in automatic mode:
mem_search({ query: "database migration decisions", mode: "auto" })
```

— Full mechanism, exact limits, and a runnable example. v13.35.0

02 Native memory hook bridge via PreToolUse NEW

90

A `PreToolUse` hook now intercepts native `memory_search` , local `memory` tool reads, and Read/Grep/Glob/Bash reads (`cat` , `head` , `tail` , `sed` , `grep` , `rg`) scoped to Claude Code per-project `memory` folders, a configured `autoMemoryDirectory` , the Codex `memories` folder, or configured note roots — injecting an automatic `mem_search` result (capped at 10,000 characters) as extra context while letting the native tool run normally. It can be disabled independently of other native-memory settings with `CLAUDE_MEM_MEMORY_SEARCH_HOOK_ENABLED=false` .

Disable the native memory bridge and the injected session instructions in a read-only hosted connection where local worker supplementation is unwanted.

```
$ CLAUDE_MEM_MEMORY_SEARCH_HOOK_ENABLED=false
CLAUDE_MEM_MEMORY_INSTRUCTIONS_ENABLED=false npx claude-mem
install --provider host
```

— Names every intercepted tool, scope, cap, and the disabling flag. v13.35.0

03 **Opt-in Markdown note watcher** NEW 90

`CLAUDE_MEM_MEMORY_WATCH_ROOTS` (a JSON array of up to 8 `{ "path", "project", "platformSource"? }` entries) configures folders that are polled for Markdown notes, which are imported into the mapped project; hidden files, symlinks, and files over 64 KiB are skipped.

Import a local Obsidian vault folder into a claude-mem project so the agent can search those notes across sessions.

```
$ CLAUDE_MEM_MEMORY_WATCH_ROOTS='[{"path":"~/vault/project-
notes","project":"my-project"}]' npx claude-mem install
```

— Exact schema, limits, and a runnable install command. v13.35.0

04 **Activation script installation discovery** IMPROVED 90

`scripts/activate-progressive-memory-local.mjs` now discovers Claude Code and Codex installations present (including marketplace and local cache copies), respects `CLAUDE_CONFIG_DIR` and `CODEX_HOME`, resolves the data directory via `--memory-data-dir` flag then `CLAUDE_MEM_DATA_DIR` env then `~/.claude-mem/settings.json`, defaults to a read-only plan, and supports `--apply` and `--restore` with hash validation.

— Names exact resolution order, env vars and flags. v13.35.0

05 **save_memory MCP tool** NEW 70

Adds save_memory({ text, title?, project?, metadata? }) so the worker runtime can write explicit preferences, decisions, corrections and reusable lessons into the active project, returning a short write receipt.

— Exact tool signature but no usage example given. v13.35.0

06 **SessionStart memory-use instructions** NEW 70

SessionStart context now injects short instructions telling agents to call `mem_search` before writing, keep notes concise, omit credentials, and use `work_state_write` for

to-do lists when available; this can be suppressed with `CLAUDE_MEM_MEMORY_INSTRUCTIONS_ENABLED=false`.

Disable the native memory bridge and the injected session instructions in a read-only hosted connection where local worker supplementation is unwanted.

```
$ CLAUDE_MEM_MEMORY_SEARCH_HOOK_ENABLED=false
CLAUDE_MEM_MEMORY_INSTRUCTIONS_ENABLED=false npx claude-mem
install --provider host
```

— Names the injected guidance and the disabling flag. v13.35.0

---

07 **Hosted runtime routing via CLAUDE_MEM_RUNTIME=server** NEW

65

With `CLAUDE_MEM_RUNTIME=server`, `mem_search`, `save_memory`, and the native memory bridge all point to the hosted MCP connection and `observation_add` instead of falling back to the local worker.

— Env var named but no command example of use. v13.35.0

---

08 **OpenCode plugin dual V1/V2 support** IMPROVED

60

The OpenCode plugin now exports a dual V1/V2 shape supporting OpenCode V1 (1.18.29+) and V2, reading checkout from `ctx.location.directory` and subscribing to the V2 event bus for turns, compactions, and session deletion.

— Names versions and API surface but no example. v13.35.0

THINNER COVERAGE BELOW

09 **Readable MCP tool replies** IMPROVED

45

Replies for search, context, selected records, corpus build/list/prime/query, server observations, events, and job lookups now return concise readable text instead of raw JSON envelopes or duplicated `structuredContent`.

— Lists affected tools but no mechanism or example. v13.35.0

---

10 **Retrieval excluded from observation capture** IMPROVED

45

Calls to `mem_search`, context tools, and native `memory_*` lookups are now excluded from observation capture, so retrieval no longer generates memory entries about itself searching.

— Clear behavior change, no action needed to use it. v13.35.0

WAS THIS USEFUL?  

Graphify Labs Graphify

Feedback ⓘ v0.9.82 NOTES ↗

RANK

Graphify converts code, docs, SQL schemas, configs, and PDFs into a queryable knowledge graph with explainable relationships.

> Graphify's only change this window was a small CLI quality-of-life addition: new environment variables to suppress its post-extract promotional pointer.

└──▷ **WHAT SHIPPED · 1 FEATURE** most completely described first ? what's the number?

01 Suppress Graphify Cloud pointer after extract NEW

70

`graphify extract` prints a one-time pointer to Graphify Cloud in interactive terminals; this can now be suppressed with the `GRAPHIFY_NO_TIPS` or `GRAPHIFY_NO_CTA` environment variables, e.g. `GRAPHIFY_NO_TIPS=1 graphify extract`.

Prevent the Graphify Cloud prompt from appearing in interactive developer environments where it is unwanted.

```
$ GRAPHIFY_NO_TIPS=1 graphify extract
```

— Names exact env vars and runnable command; mechanism is minimal. v0.9.82

UPDATED SUPPORTED PLATFORM Python cross-file base-class call resolution

UPDATED SUPPORTED PLATFORM TypeScript this.field class-declaration receiver typing

v0.9.81 NOTES ↗ STABLE

Adds `GRAPHIFY_AMBIGUOUS_SCALE` env var to control AMBIGUOUS-edge weighting in community detection, keeping independent clusters separate by default.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.9.81 https://github.com/Graphify-Labs/graphify.git
# already have the repo? check out this version:
$ git checkout v0.9.81
```

Adds `GRAPHIFY_AMBIGUOUS_SCALE=1.0` environment variable to restore prior behavior of including low-confidence AMBIGUOUS edges at full weight during

> community detection.

WAS THIS USEFUL? 👍 👎

OpenAI Codex CLI

Feedback ⓘ rust-v0.163.0-alpha.2 NOTES ↗

CODE ↗ RANK

OpenAI Codex CLI runs an agent in the terminal that reads, changes, and tests code in local repositories.

> Codex CLI's latest releases focus on terminal UX polish—pinnable tasks, clickable URLs, transcript navigation—alongside backend hardening around proxy DNS security, Guardian review persistence, and new git worktree and model provider capabilities.

└──▷ **WHAT SHIPPED · 9 FEATURES** most completely described first ? what's the number?

01 **Transcript copy and navigation controls** NEW 81

Users can navigate and copy transcript blocks with `/copy`, use Ctrl+Insert to copy selections, and tune mouse-wheel scrolling speed via the `tui.mouse_scroll_speed` config key.

Set a custom mouse-wheel scroll speed in the TUI transcript to speed up long log review sessions.

yaml

```yaml
tui:
  mouse_scroll_speed: 5
```

— Names exact command, key combo, and config key. rust-v0.162.0

02 **Pinnable tasks in Command Center** NEW 60

Tasks can now be pinned in the agent Command Center using the `p` key, keeping them in a shared Pinned group when supported by the server.

— Names key binding and feature but limited mechanism detail. rust-v0.162.0

THINNER COVERAGE BELOW

03 **Clickable URLs across CLI prompts** IMPROVED 55

URLs are now clickable in approval headers, questions, MCP prompts, warnings, banners, and verification prompts, including when links wrap across multiple lines.

— Describes scope of UI surfaces but no config/flag to act on. rust-v0.162.0

04 **Managed Git worktrees for local projects** NEW 53

Adds tools for creating and listing managed Git worktrees from trusted local projects, available when the worktrees feature is enabled.

— States capability and gating flag, no exact command shown. rust-v0.162.0

05 **Gzip DotSlash artifact decompression** NEW 52

Supports gzip (`gz`) DotSlash artifacts by decompressing the executable directly to its destination.

— Names format and mechanism but no CLI flag given. rust-v0.163.0-alpha.2

06 **Custom model provider capability overrides** NEW 48

Custom Responses-compatible model providers can now have live web access and remote compaction capabilities configured.

— Describes capability without config key or exact steps. rust-v0.162.0

07 **Code Mode JavaScript streaming and tool search** NEW 43

Adds JavaScript helpers for streaming promise results as they settle, plus opt-in ranked tool search in Code Mode.

— Names two additions but no API or flag specifics. rust-v0.162.0

08 **Proxy DNS hostname authorization** IMPROVED 35

Requires hostname authorization before proxy DNS lookups are performed.

— Security change described briefly without mechanism detail. rust-v0.163.0-alpha.2

09 **Guardian review failure persistence** IMPROVED 35

Persists Guardian review failures for reports across restarts.

— Brief statement with no detail on storage or format. rust-v0.163.0-alpha.2

WAS THIS USEFUL?  

Letta Code

Feedback ⓘ v0.34.8 NOTES ↗ RANK

Letta Code is a terminal coding agent with persistent memory and identity across development sessions.

> Letta Code shipped browser-based device discovery and a new Memory tool for MemFS v2 agents.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 **Memory tool for MemFS v2 agents** NEW 30

Adds a Memory tool for MemFS v2 agents, enabling memory operations within the MemFS v2 agent architecture.

— Names MemFS v2 but no operations detailed v0.34.7

02 **Browser-based device discovery** NEW 28

Adds browser-based discovery to detect a running Letta Code instance on the current device, enabling browser clients to locate a local instance automatically.

— Describes mechanism but no concrete API or flag v0.34.8

WAS THIS USEFUL? 👍 👎

All Hands AI OpenHands

Feedback ⓘ v1.26.0 NOTES ↗

RANK

OpenHands is an open-source software-development agent that writes code, runs commands, and browses the web.

> OpenHands v1.26.0 introduces a verify-openhands testing skill and CLI with a large library of mapped behaviors, alongside several smaller UX refinements across chat, onboarding, and automation review workflows.

└──▷ **WHAT SHIPPED · 7 FEATURES** most completely described first ? what's the number?

01 **verify-openhands CLI testing skill** NEW 70

Adds the `control-openhands` CLI with a `verify-openhands` skill and a full feature map for testing and verifying OpenHands behavior, then extends this with 20 more mapped behaviors and live recipes.

— Names CLI and skill but gives no runnable usage example v1.26.0

THINNER COVERAGE BELOW

02 **Suspended workspace explanation and switch** NEW 35

Explains a suspended workspace in the backend UI and offers a workspace switch option.

— UI behavior described but no exact navigation path v1.26.0

03 **Auto-advance setup guide steps in Canvas** IMPROVED 35

Opens the next setup guide step automatically in Canvas when the current step completes.

— Describes trigger and effect but no config option named v1.26.0

04 **Super Admin setup guide on enterprise onboarding** NEW 33

Shows the Super Admin setup guide on enterprise backends during onboarding.

— States when it appears but lacks further detail v1.26.0

05 **Chat nudge for empty agent responses** NEW 28

Renders an informational nudge in chat when the agent returns an empty response.

— Describes trigger but not the nudge content or config v1.26.0

06 **Narrowed 'needs review' automation filter** IMPROVED 28

Narrows the needs review status filter for automations.

— No specifics on what the narrowing changed v1.26.0

07 **Authenticated backend view helper** NEW 23

Adds an authenticated backend view helper for apps.

— Bare mention with no mechanism or API named v1.26.0

WAS THIS USEFUL? 👍 👎

GitHub Copilot CLI

Feedback ⓘ v1.0.94 NOTES ↗ RANK

GitHub Copilot CLI is a terminal-based coding agent that can plan, write, and test code in local repositories.

> GitHub Copilot CLI v1.0.94 adds Claude Haiku 5.5 as a selectable model, tightens MCP server reliability, and introduces managed-policy controls plus a smarter Assisted Permissions judge.

└──▷ **WHAT SHIPPED · 4 FEATURES** most completely described first ? what's the number?

01 **Claude Haiku 5.5 model support** NEW 60

Adds `Claude Haiku 5.5` to model selection and `--model` flag completions, for use with `copilot --model claude-haiku-5.5`.

Use Claude Haiku 5.5 for lightweight, fast tasks where you want a smaller model.

```
$ copilot --model claude-haiku-5.5
```

— Names exact model and flag, runnable example given v1.0.94

02 **Managed policy controls for Assisted Permissions** NEW 60

Managed policy can now disable Assisted Permissions and keep sessions in Manual Approval mode; the CLI shows a policy warning when startup bypass-permission flags are suppressed by managed settings, and shows update guidance when managed settings request a newer CLI version without blocking normal prompts.

— Describes behavior and scope but no config key named v1.0.94

THINNER COVERAGE BELOW

03 **MCP server reliability fixes** IMPROVED 55

MCP enable/disable now works before server discovery without starting MCP servers, and `copilot mcp add` now recovers cleanly after interrupted MCP config initialization.

— Names subcommand and behavior but no concrete steps v1.0.94

04 **Shell-code-aware Assisted Permissions judge** IMPROVED 40

Assisted Permissions now sends visible shell code to the permission judge, reducing unnecessary manual approval prompts.

— Explains mechanism and benefit but no interface detail v1.0.94

WAS THIS USEFUL? 👍 👎

Anthropic Claude Code

Feedback ⓘ v2.1.295 NOTES ↗

CODE ↗ RANK

Claude Code is Anthropic's terminal coding agent that plans, edits, and tests code in local repositories.

> Claude Code v2.1.295 adds a hook failure-blocking mode, expands the Claude apps gateway with per-upstream model routing, TTFB timeouts and request tracing, and introduces terminal status reporting via OSC 7501 alongside new mod APIs and plugin CLI polish.

└──▷ **WHAT SHIPPED · 7 FEATURES** most completely described first ? what's the number?

01 Gateway upstream routing, timeouts and tracing NEW

90

The Claude apps gateway gains several upstream-management additions: `timeouts.upstream_ttfb_ms` caps how long a stream may take to start before failing over or returning a 502 for Bedrock, Vertex, Foundry and other cloud upstreams; an optional `models` list (with `*` wildcard support) restricts an upstream to only the listed models, including on failover; `forceLoginMethod: "gateway"` and `forceLoginGatewayUrl` user settings let `/login` open on a specific gateway on machines without managed settings; the `inference` audit event now carries an `upstream_request_id` field from Bedrock, the Anthropic API, or another upstream for support tracing; and successful inference responses include a `request-id` header so `request_id` in Claude Code telemetry matches the gateway audit log.

Route only claude-3-5-sonnet-* models to a Bedrock upstream, with a 10-second TTFB deadline before failover.

yaml

```yaml
upstreams:
  - type: bedrock
    region: us-east-1
    models:
      - "claude-3-5-sonnet-*"
    timeouts:
      upstream_ttfb_ms: 10000
```

— Multiple named config keys plus a runnable YAML example v2.1.295

02 Retry watchdog max wait environment variable NEW

75

`CLAUDE_CODE_RETRY_WATCHDOG_MAX_WAIT_MS` caps how long unattended retry mode (`CLAUDE_CODE_RETRY_WATCHDOG`) waits out 429 and 529 errors before giving up.

Cap unattended retry wait time so a CI job does not stall indefinitely on rate-limit errors.

```
$ CLAUDE_CODE_RETRY_WATCHDOG=1
CLAUDE_CODE_RETRY_WATCHDOG_MAX_WAIT_MS=60000 claude -p 'Run the
full test suite and summarise failures'
```

— Env var named with a runnable shell example v2.1.295

03 **onFailure block option for hooks** NEW 70

Command and HTTP hooks can now be configured with `onFailure: "block"`, so a hook that fails to start, times out, or exits with an unexpected code blocks the triggering action instead of letting it through.

— Config key named with clear before/after behaviour, no example shown v2.1.295

04 **Plugin CLI warnings and validate advice** IMPROVED 70

`claude plugin install`, `enable`, `disable`, and `marketplace add` now warn when the settings file they write to does not load, and `claude plugin validate` prints advice with a pasteable install line when a plugin README lacks one, without changing the exit code even under `--strict`.

— Named commands and flag, but no sample output v2.1.295

05 **Mod SDK notifications and Button components** NEW 65

Mods gain `$.ui.notify` to raise a native notification through the user's notification channel, identifying which channel sent it, and the `Button` component now accepts children (`strings` and `Text`), enabling pressable list rows with embedded chips or dim detail text.

— API and component names given without a usage example v2.1.295

THINNER COVERAGE BELOW

06 **Program Status Protocol terminal status support** NEW 50

Claude Code now supports Program Status Protocol (OSC 7501), letting terminals that implement it display whether Claude Code is working, waiting, or done.

— Protocol named but no setup or command shown v2.1.295

07 **CLI session status messages** IMPROVED 50

`claude -p` terminal sessions now print a stderr line indicating what the process is waiting for when it stays open after its last turn, and a `Backgrounding cancelled` message appears when a turn is stopped while `←` is waiting for the current tool to finish.

— Behaviour described but no example or config shown v2.1.295

WAS THIS USEFUL? 👍 👎

Block Goose

Feedback ⓘ v1.54.0 NOTES ↗ CODE ↗ RANK

Goose is an open-source AI agent with a desktop app and CLI for coding and other automated tasks.

> Goose v1.54.0 adds an HTML session export with an interactive viewer, prompts users to pick a goose mode during first-time setup, and removes Nostr-based session sharing.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 **Nostr session sharing removed** BREAKING 70

Nostr-based session sharing has been removed: the `--nostr` and `--relay` flags on `goose session export` and the `--nostr` flag on `goose session import` are retired. JSON export/import remains available, but scripts or workflows relying on Nostr sharing will break on upgrade.

– Names exact removed flags and migration impact v1.54.0

02 **Goose mode prompt during first-time setup** NEW 65

During first-time `goose configure`, users are now prompted to select a goose mode (e.g. Auto), with the choice persisted via config.set_goose_mode().

– Names config command and mechanism but not full mode list v1.54.0

THINNER COVERAGE BELOW

03 **HTML session export with interactive viewer** NEW 20

Goose now supports exporting a session to HTML with an interactive viewer, in addition to existing export options.

– Only mentioned in summary, no command or flag given v1.54.0

└──▷ **BREAKING ON UPGRADE**

! The `--nostr` and `--relay` flags on `goose session export` and the `--nostr` flag on `goose session import` are removed; any scripts or workflows using Nostr session sharing will break on upgrade.

WAS THIS USEFUL? 👍 👎

Charm Crush

Feedback ⓘ v0.98.0 NOTES ↗ CODE ↗ RANK

Terminal-based AI coding assistant that integrates LLMs, LSPs, and MCP servers into your development workflow.

> Crush v0.98.0 adds terminal status reporting via a new protocol and OAuth login for xAI's Grok models.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 **OSC 7501 Program Status Protocol support** NEW 45

Crush now supports the `OSC 7501` Program Status Protocol, enabling terminal status reporting via Rex and TUIOS.

— Names protocol and integrations but no usage steps v0.98.0

02 **Grok OAuth login for xAI** NEW 35

Adds Grok OAuth login for authenticating with xAI models directly from Crush.

— States feature but no command or flag given v0.98.0

WAS THIS USEFUL? 👍 👎

◆ Model Training & Fine-Tuning

Hugging Face TRL

Feedback ⓘ v1.15.0 NOTES ↗ RANK

TRL trains transformer language models with reinforcement learning, covering supervised fine-tuning, reward modeling, PPO, DPO, and GRPO.

> TRL v1.15.0 enables a fused LM head by default across SFT, DPO, KTO, GRPO and RLOO trainers, cutting peak VRAM 52–82% at 8k tokens and permitting up to 6.9× longer sequences, alongside selective activation checkpointing, VLM SFT improvements, async distillation/GRPO additions, vLLM integration updates, and several breaking/deprecation changes including a Python 3.10 drop and `use_liger_kernel` deprecation.

└──▷ **WHAT SHIPPED · 14 FEATURES** most completely described first ? what's the number?

01 Selective activation checkpointing in SFT NEW 95

Adds `gradient_checkpointing_kwargs={'selective': True}` to SFTConfig for selective activation checkpointing, which saves the attention output during the forward pass to recover most checkpointing slowdown at long context without requiring `torch.compile`.

Enable selective activation checkpointing in SFT to recover throughput at long context without recomputing attention outputs in the backward pass.

python

```python
from trl import SFTTrainer, SFTConfig

trainer = SFTTrainer(
    model="Qwen/Qwen2.5-0.5B",
    train_dataset=dataset,
    args=SFTConfig(
        gradient_checkpointing=True,
        gradient_checkpointing_kwargs={"selective": True},
    ),
)
trainer.train()
```

— Named config key with runnable example code v1.15.0

02 assistant_only_loss for vision datasets NEW 90

Enables `assistant_only_loss=True` in SFTConfig for vision datasets, masking non-assistant tokens in labels when using a processor's `apply_chat_template` (requires `transformers` 5.18).

Train a vision-language model with loss applied only to assistant turns, now supported on image datasets.

```python
from trl import SFTTrainer, SFTConfig

trainer = SFTTrainer(
    model="Qwen/Qwen2.5-VL-3B-Instruct",
    train_dataset=vision_dataset,
    args=SFTConfig(assistant_only_loss=True),
)
trainer.train()
```

— Named flag, version requirement and runnable example v1.15.0

03 **Fused LM head enabled by default** IMPROVED 80

A fused LM head is now enabled by default in the SFT, DPO, KTO, GRPO, RLOO, and Distillation trainers, reducing peak VRAM by 52–82% at 8k tokens and supporting up to 6.9× longer sequences on the same GPU, compatible with DeepSpeed and PEFT.

— Mechanism and numbers given, but no example command shown v1.15.0

04 **use_liger_kernel deprecated for model_init_kwargs** DEPRECATED 80

`use_liger_kernel=True` is deprecated in `DPOTrainer`, `KTOTrainer`, `GRPOTrainer`, and `RLOOTrainer` and will be removed in v2.0.0; use `model_init_kwargs={'use_kernels': True}` instead. Setting `fused_linear_cross_entropy` explicitly now raises.

— Names old and new flags with exact replacement v1.15.0

05 **PEFT adapter on lm_head now raises** BREAKING 75

A PEFT adapter on `lm_head` now raises; use `modules_to_save=['lm_head']` instead.

— Clear before/after with exact replacement config v1.15.0

06 **compute_metrics deprecated in DPO/KTO** DEPRECATED 70

In DPO and KTO, `compute_metrics` and compute_loss(..., return_outputs=True) still receive full logits via an extra forward pass but are deprecated and will be removed in v2.0.0.

— Explains mechanism and removal timeline, no replacement shown v1.15.0

07 **vLLM integration updates** IMPROVED 60

`VLLMClient` now accepts an IPv6 `base_url`, adds support for vLLM 0.31.0, and installs vLLM in the TRL Docker images by default.

— Names surfaces but no code example v1.15.0

08 **Removed and dropped support in v1.15.0** BREAKING 60

Python 3.10 is no longer supported; `nn.DataParallel` multi-GPU mode is refused (the multi-GPU slow test job is also dropped); the experimental MiniLLM trainer is removed; vLLM versions 0.20.0, 0.20.1, and 0.20.2 are no longer supported; and `prepare_multimodal_messages_vllm` is removed.

— Lists removed surfaces but no migration mechanism v1.15.0

THINNER COVERAGE BELOW

09 **VLM SFT collator processor_kwargs** NEW 50

Adds a `processor_kwargs` parameter to the VLM SFT collator to pass processor-specific arguments at collation time.

— Named config key but no example of use v1.15.0

10 **Turn-by-turn completions logging** IMPROVED 50

Renders prompts and completions in the completions logging table as raw conversation lists turn by turn instead of flat `batch_decode`'d strings, improving readability for multi-turn, tool-calling, and VLM setups.

.

— Describes behavior change with screenshot, no config shown v1.15.0

11 **AsyncGRPO for OpenEnv harnesses** NEW 45

Adds AsyncGRPO support for training OpenEnv harnesses from validated token captures.

— States mechanism briefly, no usage example v1.15.0

12 **PEFT support in AsyncDistillationTrainer** NEW 35

Adds PEFT support to `AsyncDistillationTrainer`.

— Bare name, no mechanism or example v1.15.0

13 **loss_is_scaled_for_ga declaration** NEW 35

Adds a `loss_is_scaled_for_ga` declaration for gradient accumulation loss scaling in trainers.

— Named flag but no explanation of effect v1.15.0

14 **SmolVLM chat template** NEW 30

Adds SmolVLM original and training chat template with generation markers.

— Bare feature name, no further detail v1.15.0

└──▷ **BREAKING ON UPGRADE**

! Python 3.10 is no longer supported.

! `nn.DataParallel` multi-GPU mode is refused; the multi-GPU slow test job is also dropped.

! The experimental MiniLLM trainer is removed.

! `use_liger_kernel=True` is deprecated in `DPOTrainer`, `KTOTrainer`, `GRPOTrainer`, and `RLOOTrainer` and will be removed in v2.0.0; use `model_init_kwargs={'use_kernels': True}` instead. Setting `fused_linear_cross_entropy` explicitly now raises.

! vLLM versions 0.20.0, 0.20.1, and 0.20.2 are no longer supported.

! A PEFT adapter on `lm_head` now raises; use `modules_to_save=['lm_head']` instead.

! In DPO and KTO, `compute_metrics` and compute_loss(..., return_outputs=True) still receive full logits via an extra forward pass but are deprecated and will be removed in v2.0.0.

! `prepare_multimodal_messages_vllm` is removed.

WAS THIS USEFUL? 👍 👎

AGENT

◆ MCP Servers & Tooling

Prefect FastMCP

Feedback ⓘ v4.1.0 NOTES ↗ RANK

FastMCP is a Python framework for building MCP servers and clients, handling transport, schema generation, authentication, and server composition.

> FastMCP 4.1.0 adds configurable proxy header forwarding, in-memory upload retention, encrypted CLI OAuth token persistence, and TokenCache support for Google auth, while also shipping a cluster of breaking changes covering MultiAuth ownership scoping, a new regex engine for tool search, path-traversal protections for skill files and OpenAPI parameters, a 30-minute HTTP session idle timeout, and a Monty 1.1 requirement for CodeMode.

└──▷ **WHAT SHIPPED · 10 FEATURES** most completely described first ? what's the number?

01 **CodeMode requires Monty 1.1** BREAKING 75

`CodeMode` requires Monty 1.1, which renames `max_duration_secs` to `max_feed_duration_secs`.

— Exact version requirement and flag rename for migration v4.1.0

02 **MultiAuth client ID scoping by source** BREAKING 65

`MultiAuth` now qualifies client IDs by source — existing MultiAuth sessions and tasks start new ownership scopes; finish pending tasks before upgrading.

— Gives before/after and a clear upgrade instruction v4.1.0

03 **Path traversal protections for skill files and OpenAPI parameters** 65

BREAKING

Skill files must resolve inside their configured directories — paths that escape those directories are now rejected. OpenAPI path parameters reject `.` and `..` segments.

— Names both surfaces and the exact rejected path segments v4.1.0

04 **Regex tool search engine change** BREAKING 60

Regex tool search now uses Pydantic's regex engine, which does not support lookarounds or backreferences.

— Names the engine and the unsupported regex features v4.1.0

05 **HTTP session idle timeout** 60

BREAKING

HTTP sessions now expire after the MCP SDK's 30-minute idle default instead of never.

— States exact default and prior behavior v4.1.0

THINNER COVERAGE BELOW

06 **TokenCache for Google auth** NEW 55

Adds `TokenCache` support to `GoogleTokenVerifier` and `GoogleProvider`.

— Names the exact classes gaining the capability v4.1.0

07 **Encrypted CLI OAuth token persistence** NEW 40

Persists encrypted CLI OAuth tokens across sessions.

— Describes behavior but not the storage mechanism v4.1.0

08 **Configurable proxy header forwarding** IMPROVED 30

Makes proxy HTTP header forwarding configurable.

— Names the capability but no flag or config key given v4.1.0

09 **Configurable upload retention** NEW 30

Adds configurable upload retention in memory.

— States the addition without a config key or limit v4.1.0

10 **Configurable token-introspection audiences** NEW 25

FastMCP 4.1.0 adds configurable token-introspection audiences.

— Mentioned only in summary, no mechanism given v4.1.0

└──▷ **BREAKING ON UPGRADE**

! `MultiAuth` now qualifies client IDs by source — existing MultiAuth sessions and tasks start new ownership scopes; finish pending tasks before upgrading.

! Regex tool search now uses Pydantic's regex engine, which does not support lookarounds or backreferences.

! Skill files must resolve inside their configured directories — paths that escape those directories are now rejected.

! OpenAPI path parameters reject `.` and `..` segments.

! HTTP sessions now expire after the MCP SDK's 30-minute idle default instead of never.

! `CodeMode` requires Monty 1.1, which renames `max_duration_secs` to `max_feed_duration_secs`.

WAS THIS USEFUL?  

hayhooks

Feedback ⓘ v2.0.0 NOTES ↗ RANK

Easily deploy Haystack pipelines as REST APIs and MCP Tools.

> Hayhooks 2.0 introduces a full durable-execution engine — Redis-checkpointed pipelines with human-in-the-loop and reattachable SSE streaming — while v1.25.0 adds opt-in HTTP header injection into pipeline wrappers.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 **Durable execution engine for pipelines** NEW 100

Install support via `pip install 'hayhooks[durable]'` (requires Haystack 3.1+ and Redis 6.2+ or Valkey 7.2+) and enable it with `HAYHOOKS_DURABLE_MODE=true`; wrappers implementing `run_durable` / `run_durable_async` automatically gain submit, inspect, cancel, resume, and stream routes. Default (live) mode now rejects durable wrappers, and in durable mode `hayhooks run --workers` must be `1`, scaling instead via replicas. Redis connection, capacity, payload/stream limits, leases, retries, and shutdown handoff are configured via `HAYHOOKS_DURABLE_*` environment variables, and a new `GET /status` endpoint reports `durable_mode` and durable store health. Submitting a durable run returns `202 Accepted` with an execution ID and links to inspect, cancel, resume, and stream while it continues in the background; runs support human-in-the-loop via `suspend` with a public wait reason and typed, validated resume input defined by `durable_resume_model`, plus idempotent submission via a caller-supplied idempotency key. SSE streams reattach via `Last-Event-ID` and can be served by any replica, delivering chunks as workers flush them and ending with the authoritative result. The engine can also be embedded directly, without the Hayhooks server, via `DurableDeployment`, `DurableRuntime`, `RedisExecutionStore`, `MemoryExecutionStore`, and `create_durable_router` in any asyncio or FastAPI app, with `hayhooks.durable` requiring only Pydantic and Loguru until FastAPI, Haystack, or Redis are actually used.

Implement a durable agent wrapper with human-in-the-loop approval that checkpoints across restarts.

```python
from pydantic import BaseModel
from hayhooks import BasePipelineWrapper, DurableContext

class Task(BaseModel):
    question: str

class Approval(BaseModel):
    approved: bool

class Report(BaseModel):
    answer: str

class PipelineWrapper(BasePipelineWrapper):
    durable_revision = 'research-agent-v1'
    durable_resume_model = Approval

    def setup(self) -> None:
        # set up self.pipeline here
        pass

    async def run_durable_async(self, context: DurableContext,
task: Task) -> Report:
        if not context.state.get('approved'):
            decision = context.resume_input
            if decision is None:
                await context.suspend({'kind': 'approval',
'message': f'Research {task.question!r}?'})
            if not Approval.model_validate(decision).approved:
                raise ValueError('research was rejected')
            context.state['approved'] = True
        # pipeline runs from last checkpoint
        return Report(answer='...')
```

Embed the durable execution engine in an existing FastAPI app without running the full Hayhooks server.

```python
from fastapi import FastAPI
from hayhooks.durable import DurableDeployment, DurableRuntime,
RedisExecutionStore, create_durable_router

store = RedisExecutionStore(url='redis://localhost:6379')
runtime = DurableRuntime(store=store)
deployment = DurableDeployment(runtime=runtime, wrappers=
[MyPipelineWrapper()])
router = create_durable_router(deployment)

app = FastAPI()
app.include_router(router)
```

— Names install command, env vars, endpoint, and embedding classes precisely. v2.0.0

02 **Typed HTTP header injection in pipeline wrappers** NEW

95

Pipeline wrappers can now receive caller HTTP headers by declaring `headers: Headers | None = None` (importing `Headers` from `starlette.datastructures`) in `run_api`, `run_chat_completion`, `run_response`, and their `async` variants. Injected `headers` are excluded from JSON/multipart request bodies, OpenAPI/MCP input schemas, `/run` payload logs, and trace tags to prevent spoofing and leakage, while A2A, MCP, and direct non-HTTP calls leave `headers` as `None`, requiring wrappers to handle missing credentials explicitly. Requires `fastapi-openai-compat>=1.4.0`.

Forward a per-request API key from the caller to a downstream service, without exposing it as a pipeline input field.

```python
from starlette.datastructures import Headers
from hayhooks import BasePipelineWrapper

class PipelineWrapper(BasePipelineWrapper):
    def setup(self) -> None:
        pass  # set up your pipeline/agent here

    async def run_api_async(self, question: str, headers: Headers
| None = None) -> str:
        api_key = headers.get('x-api-key') if headers else None
        if not api_key:
            raise ValueError('Missing x-api-key header')
        # forward api_key to your downstream component
        ...
```

Forward caller credentials in a streaming chat completion handler so each user's request uses their own token.

```python
from typing import AsyncGenerator
from starlette.datastructures import Headers
from hayhooks import BasePipelineWrapper,
async_streaming_generator

class PipelineWrapper(BasePipelineWrapper):
    def setup(self) -> None:
        pass  # set up your agent here

    async def run_chat_completion_async(
        self,
        model: str,
        messages: list[dict],
        body: dict,
        headers: Headers | None = None,
    ) -> AsyncGenerator[str, None]:
        token = headers.get('authorization') if headers else None
        # use token when constructing pipeline run args
        return async_streaming_generator(
            pipeline=self.agent,
            pipeline_run_args={'messages': messages, 'api_key':
token},
        )
```

— Exact parameter, import, and version requirement given with runnable examples. v1.25.0

THINNER COVERAGE BELOW

03 Explicit registry argument required in server internals

40

BREAKING

Several internal server helper functions in `hayhooks.server.utils` covering MCP, A2A, and module loading now require an explicit pipeline registry argument, breaking code that called these internals directly.

— Names affected module and areas but not the specific functions changed. v2.0.0

└──▷ **BREAKING ON UPGRADE**

! Default (live) mode now rejects durable wrappers — serve them with `HAYHOOKS_DURABLE_MODE=true`.

! In durable mode, `hayhooks run --workers` must be `1`; scale out with replicas instead.

! Several internal server helpers in `hayhooks.server.utils` (MCP, A2A, and module loading) now require an explicit pipeline registry argument.

WAS THIS USEFUL? 👍 👎

◆ AI Agent Frameworks

Vercel AI SDK

Feedback ⓘ ai@7.0.135 NOTES ↗ RANK

The Vercel AI SDK provides TypeScript APIs for model generation, structured output, tool use, and streaming application interfaces.

> Vercel AI SDK 7.0.135 adds a persistent WebSocket chat transport for streaming applications and a top-level `dimensions` setting for embedding functions.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 **Persistent WebSocket chat transport** NEW 62

A new persistent WebSocket chat transport supports correlated streaming, serialized backpressure, cancellation, validated protocol frames, explicit lifecycle management, and sequence-based resume support.

— Mechanism described but no API name or usage example given ai@7.0.135

THINNER COVERAGE BELOW

02 **Dimensions setting for embeddings** NEW 35

A top-level `dimensions` setting is now available for embedding functions, letting callers configure embedding output size.

— Named config key but no mechanism or usage example ai@7.0.135

WAS THIS USEFUL? 👍 👎

LangChain

Feedback ⓘ langchain-core==1.6.9 NOTES ↗ RANK

LangChain is an open-source framework that orchestrates applications powered by language models.

> LangChain shipped two small but distinct additions this window: dynamic retry logic in langchain-core and OpenAI Decisions API support in langchain-openai.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 **Callable support in with_retry** IMPROVED 65

`with_retry` now accepts a callable, enabling dynamic retry behavior computed at runtime rather than a static configuration.

Pass a callable to `with_retry` to dynamically compute retry configuration based on runtime context.

python

```
from langchain_core.runnables import RunnableLambda

chain = RunnableLambda(lambda x:
x).with_retry(retry_if_exception_type=lambda exc: isinstance(exc,
ValueError))
```

— Names exact method and shows runnable example. langchain-core==1.6.9

THINNER COVERAGE BELOW

02 **OpenAI Decisions API support** NEW 30

langchain-openai adds support for the OpenAI Decisions API.

— Bare mention, no mechanism or usage shown. langchain-openai==1.7.0

WAS THIS USEFUL? 👍 👎

Agno (formerly Phidata) Feedback ⓘ v3.1.2 NOTES ↗

Agno is an agent framework and runtime that orchestrates and runs multi-agent systems.

Agno v3.1.2 adds conversation compaction with a searchable archive, HyDE query transforms for RAG, a unified browser origin policy for AgentOS, and configurable follow-up generation.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v3.1.2 https://github.com/agno-agi/agno.git
# already have the repo? check out this version:
$ git checkout v3.1.2
```

Adds Agent(compaction=True) for automatic conversation compaction: older turns are folded into a summary stored in a new `agno_compactions` table as a `CompactionRecord`, leaving originals intact and restorable; tune with Compaction(model=..., compact_at_tokens=..., uncompacted_runs=...), trigger on demand with agent.compact() / acompact(), inspect via `run.compaction`, and enable `searchable=True` to expose a `search_compacted_history` regex-grep tool over the archived transcript.

Adds `QueryTransformer` as a new retrieval hook and `HyDE` as its first implementation: Knowledge(vector_db=..., query_transformer=HyDE()) searches with a hypothetical answer instead of the raw question while rerankers still score against the original query.

Adds AgentOS(cors=CORSConfig(origins=..., origin_regex=..., merge_base_app=...)) and opt-in PublicSurface(enforce_browser_origins=True) to unify browser origin policy across CORS preflights, public run and cancel admission, auth error headers, workflow WebSockets, and MCP aliases.

Adds `followups: bool | FollowupConfig` on Agent and Team; `FollowupConfig` sets count range via `num_followups` or `min_followups` / `max_followups`, a separate model, and custom instructions scoped only to the follow-up call.

NEW INTEGRATION Heabsy model provider

NEW INTEGRATION FlexAI model provider

NEW INTEGRATION CodexAgent — OpenAI Codex external agent adapter

NEW INTEGRATION SixtyDBTools — 60db speech API toolkit

UPDATED SUPPORTED PLATFORM GPT-6 / GPT-5 reasoning model recognition in OpenAIResponses and OpenRouterResponses

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Local LLM Runtimes

LM Studio

Feedback ⓘ changelog · Oct 9, 2026 CHANGELOG ↗ RANK

LM Studio is a desktop application that runs local LLMs and provides model discovery, downloads, chat, and a local API server.

> LM Studio 0.4.26 adds two new local API endpoints, enables simultaneous operation with Bionic, and extends CUDA 13 support to Windows ARM.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 New compatible API endpoints for local inference NEW

70

Adds `/v1/systemone`, a TypeSafe AI compatible endpoint, and `/v1/decisions`, an OpenAI compatible endpoint, both for local model inference.

— Names both endpoints but not their request/response shape. changelog-20261009-d52f0792

02 Bionic coexistence support NEW

60

Enables coexistence with Bionic (1.1.8+) so both apps can run simultaneously, with Bionic delegating local inference to LM Studio.

— Explains delegation mechanism but no setup steps. changelog-20261009-d52f0792

THINNER COVERAGE BELOW

03 CUDA 13 support on Windows ARM NEW

30

Adds CUDA 13 support on Windows ARM.

— Bare statement with no further detail. changelog-20261009-d52f0792

WAS THIS USEFUL? 👍 👎

llmfit

Feedback ⓘ v1.1.17 NOTES ↗ CODE ↗ RANK

Hundreds of models & providers. One command to find what runs on your hardware.

> llmfit v1.1.17 adds CLI flags for percentage-based hardware headroom control and mouse support in its TUI.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 **Memory and RAM percentage caps via CLI** NEW 83

New global CLI flags `--memory-percent <PERCENT>` and `--ram-percent <PERCENT>` cap GPU VRAM and system RAM as a percentage of detected capacity (e.g. `90`, `87.5`), letting users set tighter headroom without calculating absolute sizes.

Simulate headroom-constrained hardware — e.g. reserve 10% of VRAM and 20% of RAM for the OS — before committing to a model deployment.

```
$ llmfit recommend --memory-percent 90 --ram-percent 80 --json --limit 5
```

— Named flags with example values and runnable command v1.1.17

THINNER COVERAGE BELOW

02 **Mouse support in TUI** IMPROVED 33

The TUI now supports mouse input, enabling click-based navigation alongside existing keyboard bindings.

— Described briefly with no specific UI detail v1.1.17

WAS THIS USEFUL? 👍 👎

Ollama

Feedback ⓘ v0.40.2 NOTES ↗ CODE ↗ RANK

Local large language model runner that downloads, serves, and chats with models through a CLI and REST API.

> Ollama v0.40.2 automatically migrates existing models to the llama.cpp runner in the background and fixes duplicate `ollama list` entries, while `ollama launch claude` now passes the model's full context window to Claude Code.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 Automatic model upgrade to llama.cpp runner IMPROVED

85

Models downloaded with earlier versions of Ollama are automatically upgraded in the background on first run for better performance and compatibility with llama.cpp; `ollama list` no longer shows duplicate entries for models upgraded this way. A shell one-liner against `localhost:11434/api/tags` and `localhost:11434/api/show` can be piped through `jq` to find and remove backed-up pre-upgrade model blobs via `ollama rm`.

Remove backed-up pre-upgrade model blobs to reclaim disk space after Ollama has upgraded your models to llama.cpp format.

```
$ curl -s localhost:11434/api/tags | jq -r '.models[].name' |
while read -r m; do
  curl -s localhost:11434/api/show -d "{\"model\":\"$m\"}" | jq -
r \
    'select(any(.manifests[]?; .runner == "llamacpp")) |
.manifests[] | select(.runner == "ggml") | .digest'
done | sort -u | xargs -n1 ollama rm
```

— Names endpoints, flags and includes a runnable cleanup command v0.40.2

02 Full context window passed to Claude Code launcher IMPROVED

65

`ollama launch claude` now sets `CLAUDE_CODE_MAX_CONTEXT_TOKENS` to the model's full context window, so models with large contexts (e.g. 1M tokens) no longer compact prematurely against Claude Code's 180k default.

— Names the env var and numbers but no runnable example v0.40.2

WAS THIS USEFUL?  

◆ Agentic Infrastructure

Perplexity API

Feedback ⓘ snapshot-20261009 NOTES ↗ RANK

Perplexity API Platform - build with the Router, Agent, Search, and Embeddings APIs. Real-time, web-wide research and Q&A capabilities for your products.

> Perplexity API added Claude Haiku 5.5 support to the Agent API and upgraded the Decisions API to a cheaper, newer model.

└──▷ WHAT SHIPPED · 2 FEATURES most completely described first ? what's the number?

01 Decisions API model upgraded to pplx-decider-v1.1-27b

IMPROVED 85

The Decisions API's `pplx-decider-v1-27b` model is replaced by `pplx-decider-v1.1-27b`, cutting input cost from $0.04 to $0.02 per million tokens. Existing requests that still name the old model continue to work and are billed at the new $0.02 rate.

Switch an existing Decisions API integration to pplx-decider-v1.1-27b to halve input token costs immediately.

```
$ curl https://api.perplexity.ai/decisions \
  -H 'Authorization: Bearer <YOUR_API_KEY>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "pplx-decider-v1.1-27b", "input": "
<decision_input>"}'
```

— Exact model names, pricing before/after, backward-compat behavior stated

snapshot-20261009

02 Claude Haiku 5.5 in Agent API NEW

70

The Agent API now supports the `anthropic/claude-haiku-5-5` model. Requests exceeding 100k input tokens are billed at higher rates.

Call the Agent API with Claude Haiku 5.5 for high-throughput agentic tasks that stay under 100k tokens to avoid elevated billing.

```
$ curl https://api.perplexity.ai/agent \
  -H 'Authorization: Bearer <YOUR_API_KEY>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "anthropic/claude-haiku-5-5", "messages":
[{"role": "user", "content": "<prompt>"}]}'
```

— Names model and curl call; billing threshold given but no exact rate snapshot-20261009

WAS THIS USEFUL? 👍 👎

OpenAI TypeScript SDK Feedback ⓘ v7.31.0 NOTES ↗

RANK

The OpenAI TypeScript SDK provides typed JavaScript and TypeScript clients for OpenAI APIs and streaming responses.

> The OpenAI TypeScript SDK added API support for prewarmed hosted environments in v7.31.0.

└──▷ **WHAT SHIPPED · 1 FEATURE** most completely described first ? what's the number?

01 API support for prewarmed hosted environments NEW

20

v7.31.0 adds API support for prewarmed hosted environments.

— Only a bare feature name, no mechanism or API details given v7.31.0

WAS THIS USEFUL? 👍 👎

NVIDIA NeMo Curator

Feedback ⓘ v1.4.0 NOTES ↗ RANK

NeMo Curator builds and scales data-curation pipelines for text, image, video, and audio training datasets.

> NeMo Curator v1.4.0 adds Lance-format I/O stages and GlotLID language identification, rounds out the audio pipeline with four new ASR adapters and new processing stages (sound event detection, hallucination filtering, best-transcript selection, normalization), and extends deduplication, pipeline execution, and core dependency handling.

└──▷ **WHAT SHIPPED · 12 FEATURES** most completely described first ? what's the number?

01 **Lance format I/O stages** NEW 65

Adds a `LanceReader` stage for reading Lance-format datasets into pipelines, a `LanceWriter` stage for writing pipeline output to Lance-format datasets, and an interleaved `LanceReader` stage for multimodal (interleaved) Lance dataset ingestion.

– Names the stages but no usage mechanism given v1.4.0

02 **Language identification options** NEW 65

Adds `GlotLID` language and script identification support as an alternative to FastText-based language detection, and adds an optional `language` parameter to `FastTextLangId` to override automatic language detection.

– Named classes and parameter but limited mechanism detail v1.4.0

03 **Deduplication stage improvements** IMPROVED 65

Adds support for dropping the deduplication ID field after deduplication runs, adds normalization support to exact and fuzzy deduplication stages, adds more custom metrics to the minhash stage, reduces KMeans fit memory overhead in semantic deduplication, and defaults pairwise precision to `fp16` in semantic deduplication to cut memory usage.

– Lists concrete changes including a default value but no numbers v1.4.0

04 **Four new ASR adapters for audio pipelines** NEW 60

Adds four new ASR adapters for audio transcription pipelines: Qwen-Omni (in-process), Qwen3-ASR, `NeMo FastConformer`, and Faster-Whisper.

– Four adapters named but no usage or behavior detail v1.4.0

05 **New audio processing stages** NEW 60

Adds a sound event detection inference stage and a sound event detection postprocessing stage, a rule-based ASR hallucination filter stage, a best-ASR-prediction

selection stage to pick the highest-quality transcript across multiple ASR adapters, and transcript normalization stages for audio pipelines.

— Several named stages, but behavior described only briefly v1.4.0

06 **Runtime and dependency updates** IMPROVED 60

Updates to Ray 2.57.0, Dynamo 1.3.0, and vLLM 0.22 (cu129), enabling newer GPU support and performance improvements; also promotes `datasets` to a core dependency, removing it from optional extras.

— Exact versions named but impact only generally described v1.4.0

07 **Custom vLLM init args in semantic dedupe tutorials** IMPROVED 60

Adds `vllm_init_kwargs` exposure in semantic dedupe tutorials, letting users pass custom vLLM initialization arguments to the embedding server.

— Named config key with clear purpose, limited to tutorials v1.4.0

THINNER COVERAGE BELOW

08 **Remote URL streaming for CC WARC iterator** IMPROVED 55

Adds remote URL support for the CC WARC iterator, enabling direct streaming from remote Common Crawl archives.

— Clear capability named, but no usage example given v1.4.0

09 **Ray Data pipeline execution tuning** IMPROVED 55

Adds scheduler observability for Ray Data pipelines, per-node stage worker sizing that lets operators tune concurrency independently for each node type, and support for pipeline stage inputs to accept more than one input task type.

— Three execution improvements named without configuration specifics v1.4.0

10 **PyArrow fast path for JSONL reads** IMPROVED 40

Uses a PyArrow fast path for JSONL reads, improving I/O throughput.

— States the mechanism but no benchmark or scope v1.4.0

11 **LLM-as-a-judge evaluation** NEW 35

Adds LLM-as-a-judge evaluation to the eval suite.

— Bare name with no mechanism or usage detail v1.4.0

12 **Nemotron-CLIMB data-curation recipe** NEW 35

Adds the Nemotron-CLIMB data-curation recipe.

– Bare name with no description of what the recipe does v1.4.0

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.8.142 NOTES ↗ RANK

HyperFrames is an open-source HTML-to-video renderer that runs in AI-agent workflows.

> HyperFrames added environment-variable based CLI authentication, letting host apps sign users in without persisting credentials to disk.

└──▷ **WHAT SHIPPED · 1 FEATURE** most completely described first ? what's the number?

01 **Environment variable CLI authentication** NEW 70

The CLI now reads `HEYGEN_ACCESS_TOKEN` from the environment for credential-free authentication, taking precedence over `~/.heygen/credentials` when set.

– Names exact env var and config path precedence, no usage example given v0.8.142

WAS THIS USEFUL? 👍 👎

◆ Inference Serving & Optimization

Microsoft ONNX Runtime

Feedback ⓘ v1.31.0 NOTES ↗

RANK

ONNX Runtime runs machine-learning models across hardware and operating systems with an open-source inference engine.

> ONNX Runtime's CUDA Plugin EP shipped new sparse- and paged-attention operators (SparsePagedAttention, DynamicSparseAttention, SparseAttentionIndexer), INT4 paged KV caches, expanded MatMulNBits and QMoE precision support (INT2, FP8, FP4, BF16), and a new `session.gqa_value_layout` option, alongside two breaking changes to `GatherBlockQuantized` semantics and core/plugin schema compatibility.

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

01 **MatMul auto-tuning and DeepGEMM FP8 path** NEW 90

Adds opt-in FP16/BF16 `MatMul` auto-tuning via `ep.cuda.enable_gemm_auto_tune=1` or `ORT_CUDA_GEMM_AUTO_TUNE=1` on SM90+, letting the runtime pick the fastest kernel (cuBLAS, GEMV, or tinygemm2) for a workload. Also adds an opt-in SM90 DeepGEMM path for eligible `MatMulBlockQuantizedFp8Weight` workloads, enabled with `ORT_FP8_MATMUL_DEEPGEMM=1`, which avoids FP16/BF16 weight-dequantization buffers.

Enable MatMul auto-tuning on SM90+ to let the runtime pick the fastest kernel (cuBLAS, GEMV, or tinygemm2) for your FP16/BF16 decode workload.

python

```python
import onnxruntime as ort

so = ort.SessionOptions()
so.add_session_config_entry('ep.cuda.enable_gemm_auto_tune', '1')

sess = ort.InferenceSession('model.onnx', sess_options=so,
providers=['CUDAExecutionProvider'])
```

Enable the DeepGEMM FP8 MatMul path on SM90 to skip the FP16/BF16 dequantization buffer and reduce memory traffic for block-quantized FP8 weight workloads.

```
$ ORT_FP8_MATMUL_DEEPGEMM=1 python run_inference.py
```

— Runnable code and env-var examples with named flags and mechanism.

plugin-ep-cuda/v0.3.0

02 **MatMulNBits kernel tuning and bit-width support**

NEW SUPPORTED PLATFORM NEW 85

Adds opt-in `MatMulNBits` GEMV experiments `ep.cuda.fpa_intb_gemv_wave_aware=1` (SM12x FP16/INT4 `M=8` tiles) and `ep.cuda.fpa_intb_gemv_paired_k=1` / `ORT_FPA_INTB_GEMV_PAIRED_K=1` (FP16/INT4 block-32 `M=5-8` profiling buckets), both disabled by default. Adds size-gated M-row chunking for large `MatMulNBits` workloads via `ep.cuda.matmul_nbits_m_chunk_size` or `ORT_MATMULNBITS_M_CHUNK_SIZE` (disabled by default), makes compact `fpA_intB` kernels the default build configuration with the full kernel matrix still available via `onnxruntime_USE_FPA_INTB_GEMM_FULL`, adds 2-bit `MatMulNBits` support with dequantization, specialized GEMV/batched paths and fused `fpA_intB` GEMM/GEMV kernels for FP16/BF16, and extends compact `MatMulNBits` to BF16.

Enable M-row chunking to bound peak memory for very large MatMulNBits workloads on memory-constrained GPUs.

```
$ export ORT_MATMULNBITS_M_CHUNK_SIZE=128
```

— Many named flags; one runnable env-var example given. plugin-ep-cuda/v0.3.0

plugin-ep-cuda/v0.2.0

03 **PagedAttention decode tiers and XQA expansion** IMPROVED

85

Adds a cuDNN paged SDPA decode tier to `PagedAttention` for FP16/BF16 causal workloads, disable with `ORT_ENABLE_CUDNN_FLASH_ATTENTION=0`; adds causal paged speculative XQA for FP16 query/output, INT8 KV caches, head size 128, GQA group size 6, and query widths 2-8; and expands non-causal `PagedAttention` beyond FlashAttention with portable paths for small cache pages, XQA speculative paths, head size 128 with 16/32/64-token pages, and ragged batches. Adds bidirectional `GroupQueryAttention` support and an `is_causal` attribute to `PagedAttention`, and further expands paged XQA with group-size-6 support, head size 256, FP16-cache decode at head size 256, and native block tables for 128-token pages.

Disable the cuDNN FlashAttention decode tier if you need to force PagedAttention to fall back to FlashAttention for debugging or compatibility.

```
$ ORT_ENABLE_CUDNN_FLASH_ATTENTION=0 python run_inference.py
```

— Detailed numeric specs across both releases; one disable flag demoed.

plugin-ep-cuda/v0.3.0 plugin-ep-cuda/v0.2.0

04 **QMoE precision and bit-width support** NEW 80

Adds CUDA execution for uniform INT2 and mixed INT2/INT4 FC1/FC2 QMoE combinations (2,2), (2,4), and (4,2) with per-projection weight-width attributes, and block-scaled FP8 QMoE support with routed-expert compaction alongside the existing per-expert global-scale mode. Adds BF16 routing support to CUDA MoE, enables FP4 QMoE kernels by default in CUDA builds with Windows support for Blackwell SM120, and adds an opt-in FP8 DeepGEMM decode path for eligible SM90 QMoE workloads via `ORT_QMOE_FP4_DEEPGEMM` (disabled by default; unavailable on Windows due to upstream header requirements).

Opt into the FP8 DeepGEMM decode path for SM90 QMoE inference to exploit Hopper tensor-core throughput.

```
$ export ORT_QMOE_FP4_DEEPGEMM=1
```

— Specific combos and flags named; one runnable env-var example. plugin-ep-cuda/v0.3.0 plugin-ep-cuda/v0.2.0

05 **`session.gqa_value_layout` session option** NEW 75

Adds the `session.gqa_value_layout` session option for applications using a BNHS `GroupQueryAttention` Value cache, with graph-inserted layout conversions; BNSH remains the default layout.

Switch a GQA-based model to BNHS Value cache layout so the plugin EP inserts layout conversions automatically rather than requiring pre-transposed inputs.

```python
import onnxruntime as ort

so = ort.SessionOptions()
so.add_session_config_entry('session.gqa_value_layout', 'BNHS')
sess = ort.InferenceSession('model.onnx', sess_options=so,
providers=['CUDAExecutionProvider'])
```

— Named session option with runnable code example and default behavior.

plugin-ep-cuda/v0.2.0

06 **Sparse attention operators for QSA/CSA selection** NEW 70

Adds CUDA `SparsePagedAttention` and `DynamicSparseAttention` operators for externally selected attention indices over paged and contiguous KV caches, supporting Qwen4-Exp QSA and DeepSeek V4 CSA attention patterns with optional attention sinks. Adds `SparseAttentionIndexer` for QSA token selection and CSA compressed-block selection, including rotary preparation, deterministic TopK, and cache updates, plus `PackedSparseAttentionIndexer` for packed, variable-length continuous batching with per-request metadata and selection outputs compatible with `SparsePagedAttention`.

— Named ops and mechanism described, but no usage example or flag. `plugin-ep-cuda/v0.3.0`

---

07 **`GatherBlockQuantized` out-of-range index behavior change**

60

BREAKING

`GatherBlockQuantized` out-of-range indices now produce zero output slices for both integer and floating-point quantized formats, rather than following ONNX Gather error semantics.

— Clear before/after behavior change readers must account for. `plugin-ep-cuda/v0.3.0`

THINNER COVERAGE BELOW

08 **Core/CUDA Plugin EP schema version compatibility requirement**

55

BREAKING

Core and the CUDA Plugin EP must use the same contributed-operator schemas; schema mismatches can cause incorrect execution or crashes. Building both from the same ONNX Runtime revision (matched version 1.31.0) is required — the minimum core version 1.24.4 alone does not guarantee compatibility.

— Exact version numbers given as a migration constraint. `plugin-ep-cuda/v0.3.0`

---

09 **INT4 paged KV caches with per-channel scales** NEW

50

Adds INT4 paged KV caches with per-channel scales, controlled by the `onnxruntime_USE_INT4_KV_CACHE` build option.

— Build flag named but no example of use given. `plugin-ep-cuda/v0.2.0`

---

10 **Elementwise and reduction kernel performance optimizations**

IMPROVED

50

Adds a small innermost-axis Split fast path for 2-4 outputs, eliminating per-invocation pinned-memory allocations and metadata uploads, and a single-memcpy Slice fast path for contiguous subregions. Parallelizes `ArgMax` / `ArgMin` over wide last axes, optimizes wide-last-axis `TopK`, accelerates low-lane INT64 `CumSum`, and registers BF16 `ReduceMean` kernels.

— Names specific ops and behaviors but no user-facing flag or example.

plugin-ep-cuda/v0.3.0 plugin-ep-cuda/v0.2.0

11 **EngramGate and NGramHashMapping operators** NEW 45

Adds CUDA kernels for the DeepSeek Engram contrib operators `EngramGate` and `NGramHashMapping`, then extends them for Qwen4-Exp with normalized gated-value output, autoregressive history, per-head offsets, and EOS/segment resets.

— Named ops, modest mechanism detail, no usage example. plugin-ep-cuda/v0.3.0 plugin-ep-cuda/v0.2.0

12 **Workspace-memory accounting for graph partitioning** NEW 35

Adds workspace-memory accounting and reporting for resource-constrained graph partitioning, including packed attention workspace recipes and optional-input-aware shape handling.

— Describes purpose but no concrete flag or example. plugin-ep-cuda/v0.2.0

13 **`GatedDeltaNet` operator for speculative decoding** NEW 25

Adds a compact `GatedDeltaNet` operator with transition capture for replaying accepted speculative prefixes, plus BF16 support.

— Brief description, no example or flag given. plugin-ep-cuda/v0.2.0

NEW DETECTION WebNN EP GroupNormalization, GroupNorm, SkipGroupNorm operator support

NEW SUPPORTED PLATFORM CoreML EP device-based execution provider selection via OrtEpFactory

NEW SUPPORTED PLATFORM Statically linked plugin EP support with static WebGPU plugin build

RETIRED SUPPORTED PLATFORM ACL EP deprecated

└──▷ **BREAKING ON UPGRADE**

! `GatherBlockQuantized`: out-of-range indices now produce zero output slices for both integer and floating-point quantized formats, rather than following ONNX `Gather` error semantics.

! Core and the CUDA Plugin EP must use the same contributed-operator schemas; schema mismatches can cause incorrect execution or crashes. Building both from the same ONNX Runtime revision (matched version: 1.31.0) is required — the minimum core version 1.24.4 alone does not guarantee compatibility.

v1.31.0 NOTES ↗ LATEST STABLE

ONNX Runtime 1.31.0 stabilizes model-package and EPContext APIs, adds workspace verification, in-memory compilation over 2 GB, and new contrib operators for MoE and hyper-connection architectures.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v1.31.0
https://github.com/microsoft/onnxruntime.git
# already have the repo? check out this version:
$ git checkout v1.31.0
```

Stabilizes model-package C and C++ APIs (`OrtApi::GetModelPackageApi` , `OrtModelPackageApi`) for loading packages, selecting compatible model variants, and creating sessions.

Stabilizes named EPContext data read/write callbacks with EP support negotiation so registered callbacks do not silently fall back to filesystem I/O.

Extends the Compile API to write external initializers and EP context data to buffers, enabling in-memory compilation workflows for models larger than 2 GB.

Adds `session.strict_workspace_verification=1` to fail initialization on workspace declaration overruns or nonzero orphaned reservations.

Adds `VarlenNGramHashMapping` contrib operator for packed DeepSeek Engram batching with request-local history, EOS resets, and segment boundaries.

+8 more

└──▷ **BREAKING ON UPGRADE**

- ! `GatherBlockQuantized` out-of-range indices now produce zero output slices across CPU, CUDA, and WebGPU, deliberately differing from ONNX `Gather` error semantics.
- ! The experimental EPContext callback lookup entries were removed; consumers must use `OrtApi::GetModelPackageApi` and the stable `OrtModelPackageApi` table.
- ! Native builds now default to 1DS telemetry; Windows TraceLogging must be explicitly selected with `--use_windows_telemetry` or `onnxruntime_USE_WINDOWS_TELEMETRY=ON` .

WAS THIS USEFUL? 👍 👎

◆ Inference Providers & GPU Clouds

Eigen Labs Darkbloom

Feedback ⓘ v0.9.19 NOTES ↗

CODE ↗ RANK

Private Inference Network on Idle Macs

> Darkbloom v0.9.19 adds persistent SSD write budgeting for provider caches, a new admin API for machine pins and Autopilot controls, daily Autopilot earnings floors, and Stripe withdrawal queuing to prevent failed payouts.

└──▷ WHAT SHIPPED · 4 FEATURES most completely described first ? what's the number?

01 SSD prefix-cache write budgeting NEW

95

Surfaces `DARKBLOOM_PREFIX_CACHE_DISK_GB` and `DARKBLOOM_PREFIX_CACHE_SSD_MAX_WRITE_GB_PER_DAY` environment variables, forwarded into newly installed launchd provider jobs to control SSD prefix-cache write endurance. A persistent, root-wide rolling-day SSD write ledger is enforced across cache instances, model reloads, and provider restarts; exhausted budgets retain read access and report `write_rate_limited`.

Cap daily SSD prefix-cache writes on a provider node to protect drive endurance, while keeping a fixed disk allocation for the cache.

```
$ export DARKBLOOM_PREFIX_CACHE_DISK_GB=100
export DARKBLOOM_PREFIX_CACHE_SSD_MAX_WRITE_GB_PER_DAY=20
curl -fsSL https://api.darkbloom.dev/install.sh | bash
```

— Named env vars, status code, and a runnable install command v0.9.19

THINNER COVERAGE BELOW

02 Admin API for machine pins and Autopilot controls NEW

45

Persists machine-scoped Autopilot controls behind an admin API, enabling per-machine configuration to survive restarts, and exposes admin machine pins and Always Ready diagnostics via the same new admin API surface.

— Admin API named but no endpoint paths or calls given v0.9.19

03 **Daily Autopilot earnings floors** NEW 45

Adds daily Autopilot earnings floors with cohorts and daily eligibility tracking, active from first opt-in.

– Mechanism described but no concrete floor values v0.9.19

04 **Stripe withdrawal queuing for providers** NEW 40

Queues Stripe withdrawals until payout funding is available, preventing failed payout attempts.

– Behavior described but no thresholds or API details v0.9.19

WAS THIS USEFUL? 👍 👎

Baseten

Feedback ⓘ DeepSeek V4.1 Flash Fast available on Baseten

NOTES ↗ RANK

Baseten deploys and serves machine learning models as autoscaling inference endpoints, with a model API catalog and dedicated GPU deployments.

> Baseten added DeepSeek V4.1 Flash Fast, a 552B-parameter multimodal model, to its Model APIs catalog on dedicated Fast capacity.

└──▷ **WHAT SHIPPED · 1 FEATURE** most completely described first ? what's the number?

01 **DeepSeek V4.1 Flash Fast model API** NEW 45

DeepSeek V4.1 Flash Fast, a 552B-parameter multimodal model at full precision, is now available through Baseten Model APIs on dedicated Fast capacity.

– Names model and capacity tier but no API usage details

DeepSeek V4.1 Flash Fast available on B…

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ snapshot-20261009 NOTES ↗

RANK

LangSmith provides tracing, evaluation, and deployment tools for LLM applications.

> LangSmith's biggest changes this cycle are multi-source dashboard charting with direct linking, tighter model access policy controls on the LLM Gateway, and sandbox networking/resource upgrades, alongside breaking changes capping extended trace retention at 180 days and retiring legacy permissions.

└──▷ **WHAT SHIPPED · 15 FEATURES** most completely described first ? what's the number?

01 **Redirect parameter for SSO login URL** NEW 85

Adds `redirect_to` query parameter to the organization SSO login URL (`/sso/login/<slug>?redirect_to=<path>`) so deep links land on the target page after SAML sign-in instead of the workspace root.

Send users to a specific trace page after SAML sign-in instead of the workspace root.

```
$ https://<langsmith-host>/sso/login/<slug>?
redirect_to=/o/<org>/projects/p/<project>/r/<run>
```

— Exact endpoint and example URL make it runnable snapshot-20261009

02 **Dashboard chart data sourcing and linking** NEW 83

Adds a `+ Select environment or dataset` button in agent-based workspaces to choose tracing projects, environments, or datasets as chart data sources, with Tracing and Tracing projects or environments tabs to switch between source types; pools runs across multiple selected tracing projects or environments into a single chart dataset instead of showing each separately; and enables linking directly to any dashboard from a tracing project or agent environment, with the prebuilt dashboard pre-selected by default and custom dashboards linkable via the three-dots menu. Dashboard tracing-project data sourcing is available on LangSmith Cloud US and Self-hosted querying SmithDB; dataset-backed charts are available on LangSmith Cloud EU/APAC and Self-hosted querying ClickHouse.

Link a custom dashboard to an agent environment instead of the default prebuilt dashboard.

> 📍 In your agent environment, click the three dots next to the dashboard name, then select your custom dashboard to link it.

Pool runs from multiple tracing projects into a single chart for cross-project analysis.

> 📍 In a dashboard chart, click `+ Select environment or dataset`, switch to the `Tracing projects or environments` tab, and add multiple projects — LangSmith aggregates all their runs into one dataset for the chart.

— Clear UI steps and deployment scope, no API surface `product docs`

---

03 **Model access policy controls** `IMPROVED` 78

LLM Gateway `GET /v1/models` now returns only the models permitted by the calling credential's model access policies. Model Access policies gain a new `Model configurations` provider option that allows a workspace's saved model configurations — either all or a chosen subset — so requests resolving to a configuration are no longer blocked. New model access policies now also include a generated policy name that can be kept or edited before creation.

— Names endpoint and policy option, no CLI walkthrough `snapshot-20261009`

---

04 **Agent Auth connection inspection for admins** `NEW` 60

Workspace administrators can view and inspect workspace-level Agent Auth connections — including authentication type, update details, and associated credential metadata — from the Integrations settings page.

— Names UI location, no deeper mechanism `snapshot-20261009`

---

05 **Retired repos and run-rules permissions removed** `BREAKING` 60

Built-in workspace roles no longer return the retired `repos:*` and `run-rules:*` permissions; custom roles using these permission names are rejected on creation.

— Names exact permission patterns, no migration path `snapshot-20261009`

THINNER COVERAGE BELOW

---

06 **Raw TCP access to allow-listed public IPs** `NEW` 55

Sandboxes now pass raw TCP connections to literal public IP addresses when the destination IP and port are explicitly allow-listed, supporting clients that dial a service's public IP directly.

— Explains mechanism but no config key shown `snapshot-20261009`

07 **180-day cap on extended trace retention** BREAKING 55

Traces ingested into LangSmith SaaS with extended retention are capped at 180 days; new settings above 180 days are rejected outright.

— Exact limit named, no migration steps snapshot-20261009

08 **Model price map search tool in MCP** NEW 50

Adds `list_model_price_maps` tool to LangSmith MCP for searching and paginating model pricing rules in a workspace.

— Names the tool but gives no usage example snapshot-20261009

09 **Context-aware LangSmith Chat on tracing projects** IMPROVED 50

LangSmith Chat on a tracing project now reads the active filters, view, time window, and selection; names runs and threads by what was asked; and shows its reasoning progress inline.

— Describes behavior, no commands given snapshot-20261009

10 **New revisions for install/build command updates** IMPROVED 50

Deployment API calls that update install or build commands now create a new revision; Docker sources reject these commands because they deploy prebuilt images.

— Names behavior, no endpoint given snapshot-20261009

11 **Higher open-file limit in sandboxes** IMPROVED 40

Sandboxes now use the guest kernel's maximum open-file limit, enabling highly concurrent agents and subprocesses to run without hitting descriptor limits.

— Explains benefit, no exact number given snapshot-20261009

12 **Customer IAM role assumption for ECR** NEW 35

LangSmith Cloud can now assume tagged customer IAM roles for ECR repository discovery and snapshot image pulls.

— Thin description, no setup steps given snapshot-20261009

13 **Higher file limit for Context Hub repos** IMPROVED 35

Context Hub repositories now support up to 1,500 files, up from the previous limit.

— Names new limit but not the prior value snapshot-20261009

14 **Search custom apps by name or creator** IMPROVED 30

Custom apps list now supports finding apps by name or creator.

— Minimal detail, simple UI feature snapshot-20261009

---

15 **Enterprise contract expiration warnings** NEW 25

LangSmith Cloud warns organization admins before an enterprise contract or cancelled plan expires.

— Thin, no timing or mechanism detail snapshot-20261009

└──▷ **BREAKING ON UPGRADE**

! Extended retention settings above 180 days are now applied as 180 days for new traces ingested into LangSmith SaaS; new settings above 180 days are rejected outright.

! Built-in workspace roles no longer return the retired `repos:*` and `run-rules:*` permissions; custom roles using these permission names are rejected on creation.

WAS THIS USEFUL?  

Arize Phoenix

Feedback ⓘ arize-phoenix-v20.20.0 NOTES ↗

RANK

Arize Phoenix is an open-source platform that monitors LLM application traces and evaluates their outputs.

> Phoenix v20.20.0 expands its MCP server with GraphQL tooling and a broader SQL allowlist, adds a Docker Sandboxes execution provider, and updates the Playground's model lineup.

└──▷ **WHAT SHIPPED · 4 FEATURES** most completely described first ? what's the number?

01 **GraphQL tools added to MCP server** NEW 55

The MCP server gains GraphQL schema, query, and mutation tools, enabling MCP clients to introspect and mutate Phoenix data via GraphQL.

— Names mechanism but no exact tool names or commands arize-phoenix-v20.20.0

02 **MCP SQL allowlist expanded to new tables** IMPROVED 55

The MCP SQL allowlist now includes the `annotation`, `evaluator`, `prompt`, `split`, and `job` tables, broadening what agents can query over SQL.

— Lists exact tables but no query syntax shown arize-phoenix-v20.20.0

03 **Playground model lineup update** IMPROVED 35

Adds `claude-haiku-5-5` to the Playground and removes retired Anthropic models.

— Names model but no usage or migration guidance arize-phoenix-v20.20.0

04 **Docker Sandboxes execution provider** NEW 30

Adds a Docker Sandboxes provider for sandbox execution environments.

— Bare name with no configuration detail arize-phoenix-v20.20.0

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ snapshot-20261009 NOTES ↗ RANK

Braintrust provides evaluation, tracing, and improvement workflows for AI applications.

> Braintrust shipped a wide batch of SDK, tracing and UI improvements in this window, headlined by a `span_customizers` hook for redacting and tagging spans, cooperative cancellation for JS/TS evaluations, new Roast workflow tracing, and dashboard export upgrades — alongside a breaking rename of .NET tracing APIs and removal of a Python SDK metadata field.

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

01 **.NET SDK tracing API renamed** UPDATED INTEGRATION BREAKING

85

ChatClientBuilder.UseBraintrustTracing() has been removed in the .NET SDK. Replace it with UseBraintrustLLMTracing() on `ChatClientBuilder` and move UseBraintrustFunctionTracing() to the agent builder (`AIAgentBuilder`).

— Exact migration path with named replacement APIs product docs

02 **Cooperative cancellation for JS/TS evaluations** NEW

80

Adds `signal` task hooks for cooperative cancellation in JS/TS evaluations: when an evaluation times out or its `signal` aborts, it stops starting new cases and waits for running tasks and scorers to finish. Also adds `WorkflowBatchTask` and `WorkflowBatchScorer` for workflow evaluations, and processSubmissionResult() now accepts `externalId` in place of `runId` .

Let a long-running evaluation task respect a cancellation signal so it stops cleanly on timeout.

```python
import braintrust

async def my_task(input, hooks):
    for chunk in expensive_generator(input):
        if hooks.cancel_event.is_set():
            return {'status': 'cancelled'}
        process(chunk)
    return {'status': 'done'}

braintrust.Eval(
    'my-project',
    data=dataset,
    task=my_task,
    scores=[my_scorer],
    timeout=120
)
```

— Mechanism and several named APIs, with an example `product docs`

---

03 **Span customizers in Braintrust.init** `UPDATED INTEGRATION` NEW

75

Adds `span_customizers:` argument to `Braintrust.init` to redact sensitive values or add tags and metadata to each span before the SDK exports it.

Intercept every span before export to strip a sensitive header and add an environment tag.

```python
import braintrust

def redact_and_tag(span):
    if 'authorization' in (span.metadata or {}):
        del span.metadata['authorization']
    span.tags = (span.tags or []) + ['env:production']

braintrust.init(
    project='my-project',
    span_customizers=[redact_and_tag]
)
```

— Named config argument with a runnable code example `product docs`

---

04 **Tracing for Roast workflows** NEW

65

Adds tracing for Roast workflows (`roast-ai` v1.0.0 or later), with a span per workflow run and per cog, and LLM calls of `chat` cogs nested under their cog.

– Clear span structure described, no example product docs

05 **Score configs surfaced in project_score events** IMPROVED 65

Online scoring rules and human review score configurations now appear in `project_score` events (created, updated, deleted), including pause and resume status changes.

– Names event type and lifecycle actions tracked product docs

06 **Reduced Python SDK logging overhead** IMPROVED 60

Reduced Python SDK CPU overhead when queuing and sending logs; log batches now use the full request size limit instead of half, resulting in fewer outbound requests.

– Clear before/after but no config surface named product docs

07 **Typed span context managers in Python SDK** IMPROVED 60

`with start_span(...) as span` now type-checks as Span under pyright and mypy; add_braintrust_span_processor() parameters are fully typed.

– Names exact functions affected by typing fix product docs

08 **Faster Copy for agent briefs** IMPROVED 60

Copy for agent no longer waits for trace details before copying a brief; the brief includes up to three prioritized trace IDs and CLI instructions for retrieving additional evidence.

– Names concrete limit and mechanism change product docs

THINNER COVERAGE BELOW

09 **Sub-agent token usage in agent spans** IMPROVED 50

Agent run spans now include sub-agent token usage across all models and `total_cost_usd` metadata.

– Names fields but gives no usage example product docs

10 **Point-and-click basic filter editor** IMPROVED 45

Basic filter editor now lets you edit simple point-and-click filters with their values pre-filled, rather than requiring SQL.

– UI change with clear before/after, no example product docs

11 **Inline investigation on Patterns page** NEW 45

The Patterns page (public preview) now lets you investigate a finding with Loop inline, with the selected pattern attached as context.

— Names feature and preview status, thin detail product docs

12 **Faster, fail-fast custom preprocessors** IMPROVED 40

Custom preprocessors now run faster on traces with many spans; if a preprocessor throws on any span the run fails immediately instead of silently skipping that span.

— Behavior change described without named config product docs

13 **Queued follow-up messages in Loop** IMPROVED 30

Loop queues follow-up messages while it works, so you can continue the conversation without waiting for the current turn to finish.

— Thin prose description, no named surface product docs

NEW INTEGRATION pi 1.0 codemode tracing

UPDATED INTEGRATION Claude Code subagent tracing

UPDATED INTEGRATION Anthropic MCP connector tool span naming

UPDATED INTEGRATION OpenAI Responses API and Agents API tracing

UPDATED INTEGRATION Agno agent and team run span cancellation_stage

NEW INTEGRATION AgentScope TeamPipeline.reply_stream() tracing

└──▷ **BREAKING ON UPGRADE**

! ChatClientBuilder.UseBraintrustTracing() has been removed in the .NET SDK. Replace it with UseBraintrustLLMTracing() on `ChatClientBuilder` and move UseBraintrustFunctionTracing() to the agent builder (`AIAgentBuilder`).

! In the Python SDK v0.45.0, spans no longer include `metadata.bt_daemon_version` .

snapshot-20261009 NOTES ↗ LATEST STABLE

Braintrust adds Coding Agent Insights dashboards, SQL charts, a new thread view, Loop on dashboards, dataset copy via MCP, Python/TypeScript SDK cancellation APIs, and Windows credential storage.

Adds Coding Agent Insights dashboard at `Dashboards > Coding agent insights`, showing usage by user, API key, model, tool, and branch with cost breakdowns; includes a daily Loop automation for recurring problem detection. Requires data plane v2.13.0+.

Adds a new thread view that groups each trace turn into request, agent work, and final response — tool calls appear alongside the LLM call that requested them, sub-agent work nests under its originating step, and Debugger findings are highlighted inline.

Adds SQL charts: build dashboard charts from a SQL query with a visual builder that edits the same underlying query; Braintrust injects the dashboard's timeframe, filters, and grouping at render time. Available on Pro and Enterprise plans; requires data plane v2.1.1+.

Adds Loop integration on dashboards — open Loop from an individual dashboard or chart editor to investigate data and build or refine charts in natural language, with the dashboard or chart added to context automatically.

Adds ability to copy project log rows into datasets via the Braintrust MCP server or Loop, preserving input, metadata, tags, and source log links without passing full contents through the assistant context. Requires data plane v2.18.0+.

+8 more

└─▷ **BREAKING ON UPGRADE**

! On `pi` model call spans, `reasoning_tokens` is renamed to `completion_reasoning_tokens`, `cost` is renamed to `estimated_cost`, `metadata.provider_request` is replaced by top-level fields such as `temperature` and `tools`, and skill loads now use `tool_name: 'read'` with `tool_kind: 'skill'` — update any queries using the earlier field names.

! Spans no longer include `metadata.bt_daemon_version`.

! After upgrading `bt` on Windows, downgrading to an earlier version requires logging in again, as credentials are now stored in `%USERPROFILE%\.braintrust\auth\credentials.dpapi` and are not backward-compatible.

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Open-Weight Models

DeepSeek V3.2 Exp

Feedback ⓘ price cut · Oct 8, 2026

MODEL ↗

DeepSeek V3.2 Exp is a chat model from DeepSeek.

Price cut on OpenRouter (deepseek/deepseek-v3.2): $0.28 → $0.259 in, $0.42 → $0.42 out per million tokens (-8%), first seen 8 Oct 2026.

WAS THIS USEFUL? 👍 👎

DeepSeek V4 Pro 0813

Feedback ⓘ

price increase · Oct 8, 2026 MODEL ↗

DeepSeek V4 Pro 0813 is a chat model from DeepSeek.

Price increase on OpenRouter (deepseek/deepseek-v4-pro): $0.2088 → $0.9553 in, $0.4176 → $1.9105 out per million tokens (+358%), first seen 8 Oct 2026.

Price increase on OpenRouter (deepseek/deepseek-v4-pro-0813): $0.3458 → $1.32 in, $1.0375 → $3.96 out per million tokens (+282%), first seen 8 Oct 2026.

WAS THIS USEFUL? 👍 👎

NVIDIA Nemotron 3 Nano 30B A3B

Feedback ⓘ

price increase · Oct 8, 2026 MODEL ↗

Nemotron 3 Nano 30B A3B is a chat model from NVIDIA.

Price increase on OpenRouter (nvidia/nemotron-3-nano-30b-a3b): $0.05 → $0.06 in, $0.20 → $0.24 out per million tokens (+20%), first seen 8 Oct 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3 30B A3B Feedback ⓘ

price increase · Oct 8, 2026 MODEL ↗

Qwen3 30B A3B is a chat model from Alibaba.

Price increase on OpenRouter (qwen/qwen3-30b-a3b-instruct-2507): $0.0481 → $0.10 in, $0.193 → $0.30 out per million tokens (+108%), first seen 8 Oct 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3.5-27B Feedback ⓘ price increase · Oct 8, 2026

MODEL ↗

Qwen3.5-27B is a multimodal model from Alibaba.

Price increase on OpenRouter (qwen/qwen3.5-27b): $0.195 → $0.26 in, $1.56 → $2.60 out per million tokens (+67%), first seen 8 Oct 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3.5-9B Feedback ⓘ

retirement scheduled · Oct 8, 2026 MODEL ↗

Qwen3.5-9B is a multimodal model from Alibaba.

OpenRouter lists a retirement date of 21 Oct 2026 for qwen/qwen3.5-9b, first seen 8 Oct 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3.6 27B

Feedback ⓘ price increase · Oct 8, 2026

MODEL ↗

Qwen3.6 27B is a multimodal model from Alibaba.

Price increase on OpenRouter (qwen/qwen3.6-27b): $0.32 → $0.45 in, $3.20 → $2.70 out per million tokens (+41%), first seen 8 Oct 2026.

WAS THIS USEFUL? 👍 👎

◆ Frontier Models

Mistral AI Mistral Large 4

Feedback ⓘ

context change · Oct 8, 2026 MODEL ↗

Mistral Large 4 is a multimodal model from Mistral AI.

Context window grew from 524K to 1049K tokens on OpenRouter (mistralai/mistral-large-4-0), first seen 8 Oct 2026.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · October 9, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue