THE AI TOOLCHAIN NO. 020

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED SEPTEMBER 8, 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 280 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 30 of 32 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. Amp and Hermes both stopped queuing mid-turn messages, so you can correct an agent while it works ad of waiting for a wrong answer to finish. OpenAI Agents SDK adds

vim ? help

guardrails at the individual tool and MCP-server level. Docker Sandboxes makes agent environments a declarative `sbxenv.yaml` file with plan/apply/destroy.

BUILD

Redirect an agent mid-turn instead of watching a wrong answer finish

Amp now hands messages to the agent as soon as they arrive rather than holding them until the turn completes, and Hermes pairs mid-turn steering with a full-screen subagent monitor. The wasted tokens and the re-prompt cycle after a long bad run both go away.

Amp · Hermes

GOVERN

Block a bad tool call at the tool, not after the model has already acted

Agents SDK adds input/output guardrails on individual function tools and MCP servers with customizable blocked messages, so a dangerous argument is stopped at the call boundary instead of being caught in a post-hoc trace review. ToolHive verifies skills and plugins with cosign key-signed checks and gives operator CRDs SPIFFE identities, so an unsigned artifact can't quietly enter the supply chain.

OpenAI Agents SDK · ToolHive

BUILD

Recreate an agent's sandbox from a file instead of a setup script nobody remembers writing

`sbxenv.yaml` describes the environment declaratively with parameterized args and lifecycle hooks, driven by `sbx env` plan/apply/destroy, with kits pullable from remote Git or OCI sources. Environment drift between a developer's laptop and CI stops being something you debug by comparison.

Docker Sandboxes

OPERATE

Cap agent spend per endpoint before the invoice arrives

Certiv Cost's public preview enforces per-endpoint token budgets and attributes spend to sessions, with a user-facing view in Scout; LiteLLM adds enterprise routing and security controls for RAG ingestion. Runaway loops become a rejected request rather than a month-end discovery.

Certiv

OFFENSIVE SECURITY

◆ Exploitation & C2

xalgorix

Feedback ⓘ 1 RELEASE · 2026-09-07 NOTES ↗ RANK

The xalgorix platform runs AI pentesting agents for reconnaissance, vulnerability detection, and exploitation workflows.

> xalgorix v4.6.73 hardens its scan engine with a deterministic, coverage-tracked specialist scan wave and adds a pinned CVE benchmark harness with strict recall/stability gates.

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

01 Deterministic specialist scan waves with coverage tracking

IMPROVED 50

Launches a bounded, deterministic specialist scan wave covering authorization/business logic, server-side injection, and client/source coverage. Exact endpoint/class coverage is tracked so unrelated probes cannot prematurely complete the scan plan.

— Describes mechanism and scope but no user-facing controls v4.6.73

02 Pinned CVE benchmark harness with recall/stability gates NEW

50

Adds a pinned real-world CVE benchmark with vulnerable/fixed controls and hard recall/stability gates, and keeps benchmark and agent artifacts in scan-local `tmp/` directories rather than host `/tmp`.

— Names artifact path change and gating but no runnable command v4.6.73

WAS THIS USEFUL? 👍 👎

BUILD

◆ AI Coding Agents

Replit Agent

Feedback ⓘ 1 RELEASE · SEEN 2026-09-08 NOTES ↗

RANK

Replit Agent is an AI coding agent that builds, tests, and deploys applications in Replit.

> Replit Agent's latest release focuses on deployment resilience with instant rollback and snapshot-serving, alongside editor branching, targeted search improvements, and finer-grained notification and automation controls.

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

01 **Instant rollback and snapshot-serving for deployments** NEW 60

Adds instant rollback capability enabling rapid reversion of deployments to a previous snapshot, introduces snapshot-serving in live mode so deployments can be served directly from snapshots, and adds a custom prefix capability for deployment or routing configuration.

— Names three deployment features but gives no exact commands or config keys.

snapshot-20260908

THINNER COVERAGE BELOW

02 **Per-site search delta and private page search** NEW 48

Adds per-site search delta and pauses universal search reconciliation for more targeted search behavior, and adds editor private page search enabling search within private editor pages.

— Describes search behavior changes without a UI path or command. snapshot-20260908

03 **llms.txt v2 integration for LLM crawlers** NEW 45

Adds an `llms.txt` v2 integration for exposing Repl content to LLM crawlers.

— Names the specific file format but no setup steps shown. snapshot-20260908

04 **Editor branches v2 with auto-commit** NEW 43

Introduces editor branches v2 with auto-commit support for version-controlled editing workflows.

— Names version and mechanism but no usage steps. snapshot-20260908

05 **No-publish flow option** NEW 33

Adds a no-publish flow option, allowing Repls to be configured without a public publish step.

— Describes the option but not where to configure it. snapshot-20260908

06 **Analytics v7 with user-flow tracking** NEW 32

Introduces an analytics v7 layer with user-flow analytics for tracking usage patterns.

— Version named but no metrics or access path given. snapshot-20260908

07 **PR grouping for automations** NEW 30

Adds PR grouping for automations, organizing automation runs by pull request.

— States the feature but no interface or command detail. snapshot-20260908

08 **Agent integrations settings surface** NEW 30

Introduces an agent integrations settings surface for managing third-party agent connections.

— Implies a settings UI location but no exact path. snapshot-20260908

09 **Granular notification controls** NEW 25

Adds granular notification controls, allowing finer-grained configuration of alert delivery.

— Generic description, no named settings or fields. snapshot-20260908

WAS THIS USEFUL? 👍 👎

Red Hat ripwire

Feedback ⓘ 1 RELEASE · 2026-09-08 NOTES ↗

RANK

ripwire gives coding agents a ranked, deterministic map of a repository — call graph, blast radius, and tests to run — from a CLI or over MCP.

> ripwire v0.5.0 adds Elixir as a first-class indexed language, import/dependency edges for Bash, Lua, Ruby and Elixir that change several output denominators, a churn-decay ranking mode, and dialect-aware co-change surprise detection, alongside evidence-ordered test selection and a rewrite of the MCP skill descriptions.

└──▷ WHAT SHIPPED · 8 FEATURES most completely described first ? what's the number?

01 **Import edges for Bash, Lua, Ruby and Elixir** BREAKING

85

Adds import/dependency edges (parser version 81) for Bash (`source FILE` , `. FILE`), Lua (`require`), Ruby (`require_relative` , `require` , `load`), and Elixir (`alias` , `import` , `require` , `use`), enabling `--deps` , `--arch` , and `--cochange` on codebases using those languages. This changes the `dep_files=` , `ccd` , `acd` , `nccd` (`--deps`), `propagation_cost` (`--arch`), and pair filter (`--cochange`) denominators for any corpus containing these languages — numbers recorded against an older build are not directly comparable.

— Names every syntax form and every affected flag and metric. v0.5.0

02 **churn-decay ranking mode** NEW

80

Adds `--rank-by=churn-decay` flag that emits a file-level `<recent>` block ordered by newest commit first, so queries about what changed recently are answered correctly rather than by churn frequency.

See what files changed most recently in a repo rather than what churns most often — useful for agent context before reviewing a PR.

```
$ ripwire . --rank-by=churn-decay
```

— Names exact flag and output block, includes runnable example. v0.5.0

03 **Dependency-language denominator disclosure** NEW

75

Adds `<health dep_langs=>` attribute to `--deps` output, publishing the exact set of dependency-capable languages behind the `dep_files=` , `ccd` , `acd` , `nccd` denominators so numbers are comparable across builds.

— Names the exact attribute and all four denominators. v0.5.0

---

04 **Dialect-aware co-change surprise predicate** BREAKING

75

Changes `--cochange`'s `surprising=` predicate to a per-pair question requiring a shared dependency dialect, preventing cross-dialect pairs (`.sh` ↔ `.h` , `.js` ↔ `.h` , `.py` ↔ `.cpp`) from being reported as hidden architectural debt. Pairs previously flagged as surprising across dialects (e.g. `.js` ↔ `.h`) will no longer appear.

— Explains mechanism and before/after with named examples. v0.5.0

---

05 **MCP skill description rewrite and consolidation** IMPROVED

65

Rewrites all eighteen MCP skill descriptions to fit the ~350-character client budget and folds `ripwire-efficient` into `ripwire-orient` , enforced by `test/skilldescbudgetcheck.sh` against the binary's own skill discovery.

— Names consolidated skills and the enforcing test script. v0.5.0

---

06 **Evidence-ordered tests-to-run output** NEW

60

Adds tests-to-run output in evidence order: a changed test file appears first, then its stem partner, then graph hops, with each row stating why it is included and an explicit message when the result is empty.

— Explains ordering and behaviour but no exact command shown. v0.5.0

THINNER COVERAGE BELOW

---

07 **Elixir as first-class indexed language** NEW

55

Adds Elixir as a first-class indexed language with a vendored tree-sitter grammar, call-graph extraction, and protocol implementations indexed as their own definitions.

— Describes mechanism but no command or config surface given. v0.5.0

---

08 **New gate scripts for language import checks** NEW

50

Adds five new gate scripts — `test/bashsourcecheck.sh` , `test/luarequirecheck.sh` , `test/rubyrequirecheck.sh` , `test/eliximportcheck.sh` , and `test/deplangscheck.sh` — bringing the gate total from 550 to 555.

— Lists all script names but no usage detail beyond count. v0.5.0

└──▷ BREAKING ON UPGRADE

! The `dep_files=` , `ccd` , `acd` , `nccd` (`--deps`), `propagation_cost` (`--arch`), and pair filter (`--cochange`) denominators have changed for any corpus containing Bash, Ruby, Lua, or Elixir files — numbers recorded against an older build are not directly comparable.

! The `--cochange surprising=` predicate now requires both files in a pair to share a dependency dialect; pairs that were previously flagged as surprising across dialects (e.g. `.js` ↔ `.h`) will no longer appear.

WAS THIS USEFUL? 👍 👎

herdr

Feedback ⓘ 1 RELEASE · 2026-09-07 NOTES ↗ RANK

Herdr is a coding-agent runtime that manages persistent terminal sessions across local and SSH machines.

> herdr's biggest window feature is a full multi-machine SSH management layer built around the `machine` subcommand, alongside a workspace-close safety requirement, expanded theming and pane-border controls, broadened Kitty graphics support, and an early Windows beta with agent CLI integrations.

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

01 Multi-machine SSH management via `machine` subcommand NEW

95

Adds `herdr machine add <target> --label <name>` (optionally with `--remote-session <name>` to target a named remote session), plus `herdr machine list` (with `--json` for scripting), `herdr machine rename <profile-id> --label <name>`, `herdr machine disable <profile-id>`, `herdr machine enable <profile-id>`, and `herdr machine remove <profile-id>` to manage saved profiles. Supports multi-machine connections from Linux/macOS clients to Linux/macOS servers on x86_64 and aarch64, with a combined agent list, machine-scoped navigation, notifications, and automatic reconnects; `machine add` checks remote installation capabilities and installs or updates with approval only when needed, avoiding pre-matched release versions. A `machine` token appears in sidebar rows and status-bar layouts (with conditional color support) when multiple machines are present, and an experimental `--handoff` flag on `herdr --remote` supports live server handoff during standalone setup.

Register a remote build machine under a friendly label, targeting a specific named session, so it appears in the herdr sidebar alongside Local.

```
$ herdr machine add workbox --label "Build machine" --remote-session agents
```

List all saved machine profiles with their opaque IDs in JSON for use in scripts or automation pipelines.

```
$ herdr machine list --json
```

Disable a saved machine profile without removing it, preserving credentials and label for later re-enabling.

```
$ herdr machine disable <profile-id>
```

Save a remote SSH machine profile tied to a named session so herdr reconnects automatically in the background on every client start.

```
$ herdr machine add "Build machine" --remote-session build-session
```

— Names full command set, flags, and mechanism end to end `v0.9.0`

---

02 **Custom light/dark theme color token overrides** NEW 85

`theme.custom.light.*` and `theme.custom.dark.*` config keys (`accent` , `panel_bg` , `sidebar_bg` , `active_row_bg` , `selection_bg` , `surface0` , `surface1` , `surface_dim` , `overlay0` , `overlay1` , `text` , `subtext0` , `mauve` , `green` , `yellow` , `red` , `blue` , `teal` , `peach`) override individual color tokens depending on whether `auto_switch` selects a light or dark appearance, applied after `theme.custom` ; values accept hex, named colors, rgb(r,g,b), or reset aliases. Exposed as `[theme.custom.light]` and `[theme.custom.dark]` TOML subtables layering overrides on shared custom colors.

Set distinct accent and panel colors for light vs. dark appearance when using auto_switch, so both modes reflect your team's branding.

yaml

```
theme:
  custom:
    light:
      accent: "#005f87"
      panel_bg: "#f0f4f8"
      sidebar_bg: "#e2eaf2"
    dark:
      accent: "#61afef"
      panel_bg: "#1e2228"
      sidebar_bg: "#16191e"
```

— All 19 token names and value formats listed `product docs`

---

03 **Conditional styling rules for status-bar and sidebar tokens**

NEW 85

`rules` arrays on text-valued status-bar tokens support `equals` , `contains` , `starts_with` , `gt` , and `lt` conditions with per-rule style overrides, and `ignore_case`

`= true` enables ASCII case-insensitive matching on string rules. Sidebar text and metadata tokens can likewise change color, boldness, and dimming based on their values using ordered text or numeric rules.

Color a load-average token yellow when above 60% and red above 90% — catch overloaded machines at a glance in the status bar.

toml

```
[$load]
style = { fg = '#fff' }
rules = [
  { gt = 90, style = { fg = '#f55' } },
  { gt = 60, style = { fg = '#fc0' } }
]
```

— Named rule types and config with worked example `v0.9.0`

04 **Workspace close --group now required for worktree groups** 80

BREAKING

The `--group` flag on `workspace close` (or `close_group: true` on `workspace.close`) closes a primary workspace together with its linked-worktree workspaces; omitting it returns `workspace_group_close_required`. As of v0.9.0 this flag is mandatory in that scenario — previously the whole group closed automatically when the primary workspace was closed.

Close a primary workspace that has linked-worktree workspaces open, taking the entire group down in one command instead of getting a workspace_group_close_required error.

```
$ herdr workspace close --group
```

— Named flag, error code, and before/after behavior `v0.9.0`

05 **Pane border display modes** NEW 75

`ui.pane_borders` accepts `"auto"` (borders only for split panes), `"always"` (frame even a single pane), or `"off"` (no borders), with existing boolean values still accepted; `ui.pane_outer_borders = true` is required to frame a single pane when `ui.pane_borders = "always"` is set.

Frame every pane — including a lone full-screen pane — so window boundaries are always visible.

```toml
[ui]
pane_borders = 'always'
pane_outer_borders = true
```

Always show a border even on a single unsplit pane, so the status bar and pane frame are always visible.

```toml
ui.pane_borders = "always"
```

— Config key, all three values, and runnable example `v0.9.0`

06 **Named session targeting and remote keybindings** NEW

65

Adds `--session <name>` flag to `herdr` and `herdr --remote` to target a named session instead of the default, and `--remote-keybindings server` flag to remote attach so the server's keybindings are used instead of local ones.

Connect to a remote build machine using a specific named session rather than the default, useful when the remote host runs multiple isolated herdr sessions.

```
$ herdr --remote workbox --session build-session
```

— Two flags named with a runnable example `product docs`

07 **Windows agent CLI integrations** NEW

60

Adds Windows integration support for Pi, OMP, Claude Code, Codex, GitHub Copilot CLI, Devin CLI, OpenCode, Kilo Code CLI, Droid, Kimi Code CLI, Qoder CLI, and Antigravity CLI, with unsupported install formats automatically hidden or rejected.

— Lists integrations but no config/command shown `product docs`

08 **Independent per-client terminal UI** IMPROVED

60

The terminal UI now runs in each client, keeping themes, menus, copy mode, and other presentation settings local to the viewing machine. Client updates can leave compatible servers and their running agents untouched, with missing server features disabling only the affected action instead of preventing connection.

— Describes architecture shift but no config surface `v0.9.0`

09 **Lifecycle event subscriptions start from live events** BREAKING

60

New lifecycle event subscriptions now start with live events rather than replaying retained history; API clients should subscribe before taking their initial snapshot to avoid missing changes.

— Clear behavior change with migration guidance v0.9.0

THINNER COVERAGE BELOW

10 **Additional workspace command flags** NEW 55

Adds `--trust-repository` to trust a verified repository's resolved path for that command only without modifying Git configuration, plus `--force`, `--label TEXT`, `--focus`, and `--no-focus` flags to workspace commands.

— Flags named but no usage example given product docs

11 **Muse agent detection and notification sound** NEW 55

Adds Muse agent detection for idle, working, approval, and question states, plus a `ui.sound.agents.muse` config key to override the notification sound for detected Muse agents.

— Named config key and states, no example given v0.9.0

12 **`--no-session` single-process mode removed** BREAKING 55

The `--no-session` single-process mode has been removed; all terminal UI launches now attach to a background server.

— Named flag removal with clear before/after v0.9.0

13 **Windows-native update flow** NEW 50

Windows updates install via the Windows installer, advancing the active versioned release path so new terminals and reconnected SSH sessions pick it up without requiring a server restart.

— Explains mechanism but no concrete command product docs

14 **Server upgrade required for endpoint generation compatibility** 40

BREAKING

Servers older than endpoint generation 1 require a one-time upgrade to work with updated clients.

— States requirement but no upgrade steps given v0.9.0

└──▷ BREAKING ON UPGRADE

- ! The `--no-session` single-process mode has been removed; all terminal UI launches now attach to a background server.
- ! Closing a primary workspace with open worktree workspaces now requires explicit group intent via `workspace close --group` or `workspace.close` with `close_group: true`; previously the whole group would close.
- ! New lifecycle event subscriptions now start with live events instead of replaying retained history; API clients must subscribe before taking their initial snapshot to avoid missing changes.
- ! Servers older than endpoint generation 1 require a one-time upgrade to work with updated clients.

WAS THIS USEFUL? 👍 👎

Superset

Feedback ⓘ 2 RELEASES · 2026-09-07 NOTES ↗ RANK

Superset is an agentic IDE to orchestrate 100+ coding agents in parallel. Run any agent with your own subscription.

> Superset's latest releases add a plugin marketplace with a manifest, CLI, and credential proxy, expand per-launch agent model and effort selection across the CLI, SDK, MCP, desktop, and mobile, and round out review and terminal workflows with new built-in themes, terminal script management, inline PR reply support, and diff/search improvements in the Changes pane.

└──▷ WHAT SHIPPED · 17 FEATURES most completely described first ? what's the number?

01 Agent model and effort selection across launch surfaces NEW

70

Per-launch agent models are now exposed via the `CLI`, `SDK`, and `MCP`; the desktop app adds a reasoning effort picker for the `cursor-agent`; mobile sessions let users pick the agent's model and effort when starting a session; and pinned model releases are now placed behind a submenu.

— Names each surface but lacks deeper mechanism detail. cli-v1.27.0

02 Built-in Catppuccin Latte, Solarized Light, and Vellum themes

NEW

70

Ships Catppuccin Latte, Solarized Light, and Vellum as built-in themes, selectable via Settings > Themes without installing a marketplace package.

Switch to a built-in theme such as Catppuccin Latte without installing a marketplace package.

📍 In the desktop, go to Settings › Themes and select Catppuccin Latte, Solarized Light, or Vellum from the built-in list.

— Exact navigation path and theme names given. cli-v1.27.0

03 Reply to PR review threads from Changes pane NEW

70

Enables replying to PR review threads directly from the Changes pane: open the Changes pane, locate a PR review thread, and click Reply to respond inline.

Review PR threads without leaving the desktop — open the Changes pane, find a review thread, and reply inline.

📍 In the desktop, open the Changes pane › locate a PR review thread › click Reply to respond inline.

— Exact UI steps given via usage example. cli-v1.27.0

04 **Terminal script management via CLI** NEW 65

The CLI gains a `--upsert` flag and the ability to list, edit, and delete terminal scripts via `superset cli`.

— Names the exact flag and command reader can run. cli-v1.27.0

05 **16 MB page upload cap with direct storage upload** NEW 60

Caps page uploads at 16 MB and uploads page documents straight to storage via `tRPC`, `CLI`, and `MCP`.

— Gives concrete limit and named surfaces, no steps. desktop-v1.27.0 cli-v1.27.0

THINNER COVERAGE BELOW

06 **Plugin marketplace with manifest, CLI, and credential proxy** NEW 55

Adds the plugin marketplace along with its manifest, CLI, and credential proxy, and adds a Darker theme available through the marketplace.

— Names components but no usage steps. cli-v1.27.0

07 **Diff previewing and file search in Changes/SCM pane** NEW 55

Adds image diff previewing in the SCM pane, renders binary diffs using the file pane's registry views, and adds file search to the changes sidebar.

— Names three UI additions without deeper mechanism. desktop-v1.27.0 cli-v1.27.0

08 **Device-first Workspaces page redesign** NEW 50

Adds a device-first Workspaces page with a device-on-rows layout, title, creator filter, and menu/typing performance improvements.

— Names layout elements, no navigation path given. cli-v1.27.0

09 **Admin Growth page with PostHog drilldowns** NEW 50

Adds an admin Growth page showing every growth signal with movable tiles and PostHog drilldowns.

— Names PostHog integration but no navigation path. desktop-v1.27.0

10 **Zoom shortcuts scoped to focused pane** IMPROVED 50

Scopes Cmd+/-/0 zoom to the focused terminal or browser pane.

— Names exact shortcut and scope, thin description. desktop-v1.27.0

11 **Sidebar management for projects and web dashboards** NEW 45

Adds ability to hide projects from the sidebar and delete them from the context menu, and adds a desktop-style sidebar shell to every web dashboard page.

— Describes two sidebar changes without exact UI paths. cli-v1.27.0

12 **Automations triggered on PR assignment or review request** NEW 40

Enables automations to trigger when a pull request is assigned or review-requested.

— States trigger conditions but no config or command. cli-v1.27.0

13 **'Open in' submenu for terminal link menu** IMPROVED 40

Adds an `Open in` submenu to terminal link right-clicks, then regroups the terminal link menu and renames the `Open in` destinations.

— Describes menu change but no exact destinations. cli-v1.27.0

14 **Public profiles and tier scoring on leaderboard** NEW 35

Adds public profiles, handles, achievements, and tier scoring to the leaderboard.

— Lists additions but no mechanism for scoring. cli-v1.27.0

15 **Claude/Codex agent session and subagent visibility** IMPROVED 35

Keeps a Claude session resident in the review host so the reviewer sees a pull request, and shows Claude and Codex subagents under their parent agent in the sidebar.

— Describes visibility behavior without exact UI steps. desktop-v1.27.0 cli-v1.27.0

16 **Lazy GitWatcher registration for host performance** IMPROVED 35

Lazily registers GitWatcher watchers, driven by client interest, improving host-service performance.

— Explains mechanism but gives no measurable numbers. cli-v1.27.0

17 **API-billed usage profiles in desktop app** NEW 25

Adds API-billed usage profiles in the desktop app.

— Bare mention with no mechanism or navigation. `cli-v1.27.0`

WAS THIS USEFUL? 👍 👎

sirmalloc ccstatusline

Feedback ⓘ 1 RELEASE · 2026-09-03

NOTES ↗ RANK

ccstatusline is a customizable statusline for Claude Code CLI with powerline support, themes, and widget-based configuration.

> ccstatusline v2.2.29 introduces a live Claude Status widget with incident history, unifies hide conditions across widgets, adds configurable numeric precision, and refines Git Conflicts display.

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

01 **Claude Status widget with incident history** NEW 75

Adds a Claude Status widget showing live severity along with a cached 48-hour incident-history strip. It falls back to stale data when live status can't be fetched and shows a graceful `?` when no status data is available at all.

— Describes mechanism and fallback behaviour but no config key shown. v2.2.29

02 **Unified widget hiding across widget types** IMPROVED 70

Widget hiding is unified across numeric, Git, JJ, usage, cache, and other widgets into a shared `h` checklist of supported hide conditions, with existing settings automatically migrated to the new system.

— Names the checklist and migration but not an exact command. v2.2.29

03 **Git Conflicts widget hide-on-zero and clean glyph** NEW 60

The Git Conflicts widget gains a hide-on-zero option, plus the ability to display either `⚠0` or a customizable clean glyph when the tree has no conflicts.

— Names the exact glyph and option but no config file path. v2.2.29

THINNER COVERAGE BELOW

04 **Configurable numeric precision per widget** NEW 55

Adds configurable numeric precision both per widget and globally, broken out by token, speed, percent, memory, and cost type, with an option to set explicit decimal precision for advanced configurations.

— Lists value types but no config key or command shown. v2.2.29

WAS THIS USEFUL? 👍 👎

Trailhq Graft

Feedback ⓘ 5 RELEASES · 2026-08-24 → 2026-08-31

CODE ↗ RANK

Graft builds a code knowledge graph that gives coding agents compact, repository-specific context through a CLI and MCP server.

> Graft's biggest addition this window is a `graft blast` command that seeds impact traversal from a diff and posts blast-radius results on every PR, alongside broad language expansion (Swift, R, Kotlin, Java, Dart, Clojure, PHP, Lua, Nix, plus a 14-language WASM breadth tier), new OrcaRouter and LiteLLM providers, Grok/Hermes/Antigravity host integrations, a `graft uninstall` command with convergent `init`, and opt-out telemetry via `graft telemetry disable`.

└──▷ WHAT SHIPPED · 19 FEATURES most completely described first ? what's the number?

01 --no-statusline init flag NEW

95

Adds `graft init --no-statusline` (and `GRAFT_NO_STATUSLINE=1` env var) to skip writing Claude Code's `statusLine` / `subagentStatusLine`, preserving user-defined status bars in `.claude/settings.json` or `~/.claude/settings.json`; the choice is recorded in the wiring stamp so a later session refresh cannot restore Graft's bar.

Re-initialise a repo without overwriting a custom Claude Code status bar you have already configured in `~/.claude/settings.json`.

```
$ graft init --no-statusline
```

— Exact flag, env var, config paths, and persistence mechanism named. v0.16.0

02 LLM provider integrations: OrcaRouter and LiteLLM NEW

90

Adds `orcarouter` as a first-class `GRAFT_PROVIDER` value, with `ORCAROUTER_API_KEY`, `ORCAROUTER_BASE_URL`, and `ORCAROUTER_MODEL` env-var fallbacks and `/v1/models` auto-discovery, mirroring existing `litellm` wiring; also adds a LiteLLM provider with `/v1/models` discovery as an LLM backend option.

Route all LLM calls through OrcaRouter's AI gateway instead of pointing Graft at an anonymous custom base URL.

📍 `GRAFT_PROVIDER=orcarouter` `ORCAROUTER_API_KEY`=<your-key> `ORCAROUTER_MODEL`=<model> graft build .

— Full env var names, discovery mechanism, and runnable command given. v0.16.0 v0.15.0

03 **New agent host integrations: Grok, Hermes Agent, Antigravity, Codex hook parity**

NEW 90

Adds `graft init --agents grok` to configure Grok (xAI) as a host, writing `.grok/skills/graft/SKILL.md` and a `[mcp_servers.graft]` block in `.grok/config.toml`; adds Hermes Agent host support; adds first-class Google Antigravity host support; and brings the full Claude Code hook set to Codex (hook parity).

Wire Graft into a Grok (xAI) workspace so the agent can query your codebase via MCP.

```
$ graft init --agents grok
```

— Exact files, config block, and command given for Grok; others are name-only. v0.13.0 v0.12.1

04 **Telemetry opt-out and debug subcommands** NEW 90

Adds `graft telemetry disable` and the `DO_NOT_TRACK` environment variable to opt out of anonymous usage telemetry; `graft telemetry debug` prints the exact batch that would be sent without transmitting anything.

Opt out of anonymous telemetry on a machine where `DO_NOT_TRACK` is not set and graft init has already run.

```
$ graft telemetry disable
```

Inspect exactly what telemetry your machine would send before deciding whether to opt out — nothing is transmitted.

```
$ graft telemetry debug
```

— Exact subcommands and env var with runnable examples. v0.12.1

05 **Blast-radius command for PR impact analysis** NEW 85

Adds a `graft blast` command that seeds impact traversal from changed symbols in a diff and posts the result on every PR; the `--export-viz` flag publishes the impact graph as the viewer's Context tab.

Publish a blast-radius impact graph for a PR diff so reviewers can see what else in the repo depends on the changed symbols.

```
$ graft blast --export-viz
```

— Names command, flag, and PR-posting behaviour but not the traversal algorithm. v0.12.1

06 **graft installation lifecycle: uninstall, convergent init, and auto-update**

NEW 85

Adds `graft uninstall` to fully retract every file and config block graft has ever written to a repo; updates `graft init` to retract every agent it is not about to write so re-running `init` with different `--agents` converges the repo instead of accumulating stale files; and adds auto-update for graft and the wiring it writes.

Cleanly remove graft from a repo that was wired under an older version or with different agents.

```
$ graft uninstall
```

Re-wire a repo after changing agent selection — stale agent files from prior runs are retracted automatically.

```
$ graft init --agents claude-code,cursor
```

— Two runnable commands given; auto-update mechanism left unspecified. v0.14.0 v0.12.1

07 **New `graft build` flags for indexing scope and git handling**

NEW 80

Adds `--only-dir` to `graft build` to restrict indexing to a specific directory subtree, opt-in indexing of nested git clones (previously subdirectories that were separate git repositories were skipped), and an opt-out of `.gitignore` and `.ignore` writes during `graft build`.

Limit a build to a single subdirectory — useful when only one service in a monorepo has changed and you want fast, targeted indexing.

```
$ graft build --only-dir src/auth
```

— Three flags named with a runnable example for one. v0.15.0

08 **Multi-repo and subdirectory resolution for graft init and sessions**

IMPROVED 75

`graft init` now wires every workspace child repo in addition to the parent, and `--dry-run` lists each child's writes with node/edge totals summed across children; adds `nearestGraftRoot` resolution so a session started from a subdirectory (e.g. `<repo>/src/foo/`) finds the repo's graph rather than reporting none.

— Names `--dry-run` output and resolution function, no full command shown. v0.12.1

09 **Blast-radius report detail: reviewer tagging, area naming, and source quoting**

IMPROVED 65

Blast-radius PR reports gained reviewer suggestions derived from `git log` over each affected area and scored by recency; App comments now name blast-radius areas and tag reviewers via `nameReport` then `attachOwners` in `review.ts`, closing a gap where App comments showed raw hub symbols and no `Tag:` line while CI comments on the same repo showed concept names and a reviewer; and PR comments, CLI output, and a shared helper now quote the source line where a call edge occurs.

— Named internal functions but no user-facing command or flag. v0.16.0 v0.14.0 v0.13.0

10 **Swift full-fidelity extraction** NEW 65

Adds full-fidelity (depth-tier) Swift extraction via `tree-sitter-swift` `^0.7.1`, parsing `.swift` files with complete symbol and call-edge wiring — classes, structs, enums, actors, extensions, protocols, typealiases, and top-level variables.

— Names parser version and exact symbol kinds covered. v0.15.0

11 **R language support with class system parsing** NEW 65

Adds R language support, parsing `.R` / `.r` files via `tree-sitter-r` (`npm:@davisvaughan/tree-sitter-r`), with support for R6, S4, S3 class systems and plain functions.

— Names package, file extensions, and all three class systems supported. v0.13.0

12 **Cross-language import and reference edge resolution** IMPROVED 65

The breadth tier now resolves C/C++ `#include` → file-to-file import edges, Rust `use crate::…` → file-to-module import edges, and PHP `use App\…` → file-to-class-file import edges, plus resolved reference edges (`extends`, `implements`, `new`); Java annotation usage is also wired as reference edges in the code graph.

— Names every edge pattern resolved across four languages. v0.15.0 v0.12.1

THINNER COVERAGE BELOW

13 **Cursor session scoring** NEW 55

Scores Cursor sessions using the same accounting as Claude Code sessions, tracking `graftReads`, `sourceReads`, and `savedTokens` counters in `graft/.cache/session/`.

— Named counters and cache path but no command to invoke. v0.16.0

14 **New language support: Lua, Nix, Kotlin, Java, Dart, Clojure, PHP**

NEW 55

Adds Lua language support, Nix language support, full-fidelity Kotlin extraction covering `.kt` and `.kts` files, Java language support via `tree-sitter-java` with full-fidelity extraction, Dart language support in the breadth tier, Clojure support on a broader WASM grammar bundle, and PHP language support.

— Names seven languages and two extraction tiers, no usage steps. v0.13.0 v0.12.1

15 **14-language breadth tier via generic WASM grammars** NEW 40

Adds a 14-language breadth tier via generic tree-sitter WASM grammars, covering languages without hand-written extractors.

— Gives count and mechanism but not which 14 languages. v0.12.1

16 **Opt-in LSP edge enrichment** NEW 40

Adds opt-in LSP edge enrichment (compiler-grade reference edges) layered on top of the breadth tier.

— Describes what it adds, not the opt-in mechanism or flag. v0.12.1

17 **.vue SFC indexing** NEW 40

Adds `.vue` SFC indexing via the component's `<script>` block.

— Names file type and block indexed, no further mechanism. v0.12.1

18 **GitHub App for pull request review** NEW 25

Adds a GitHub App that reviews pull requests, including those opened from forks.

— No detail on review mechanism or setup steps. v0.14.0

19 **Adjustable detail panel width in visualization viewer** IMPROVED 25

Makes the detail panel width adjustable in the visualization viewer.

— No navigation path or control described. v0.12.1

WAS THIS USEFUL? 👍 👎

Sourcegraph Amp

Feedback ⓘ 1 RELEASE · 2026-09-08 BLOG ↗

RANK

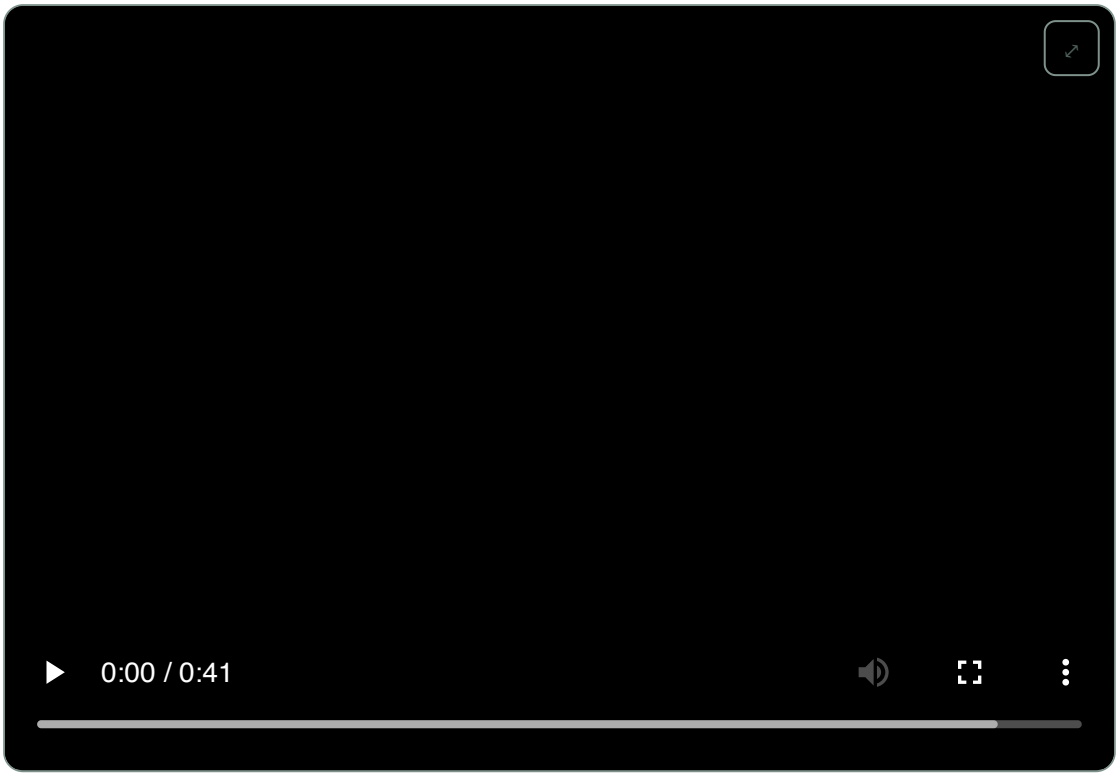
Amp is Sourcegraph's agentic coding tool for the terminal and editor, running multi-step edits with subagents and shared team threads.

> Amp now delivers mid-turn messages to the agent as soon as possible instead of queuing them until the current turn finishes, enabling real-time steering.

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

01 **Mid-turn steering of running agent** IMPROVED 65

Mid-turn messages are now delivered to the agent at the next possible opportunity, allowing real-time steering instead of waiting for the current turn to complete. Built-in actions such as Ship and Review still queue until the agent finishes, preserving end-of-turn semantics, and manual queueing remains available via Amp or the Amp CLI when immediate steering isn't wanted.



— Explains mechanism and exceptions but no exact command/flag given

Steer, Don't Queue (2026-09-08)

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

PydanticAI

Feedback ⓘ 1 RELEASE · 2026-09-08 NOTES ↗ RANK

PydanticAI is a framework that builds type-safe Python agents with dependency injection, model integrations, tools, and structured outputs.

> PydanticAI's biggest window in a while: five new model providers (Mistral, Hugging Face, Groq, Cohere, Cerebras) plus a heavily expanded xAI/Grok integration, alongside major framework additions — Monty, a sandboxed Rust-built Python interpreter for LLM-generated code; declarative Agent Specs; a full custom-capabilities and Hooks system for intercepting the agent lifecycle; a tenacity-based transport retry framework; and agent-free direct model requests.

└──▷ WHAT SHIPPED · 16 FEATURES most completely described first ? what's the number?

01 Monty: sandboxed Python interpreter for LLM-generated code NEW

98

Adds `pydantic-monty` (Python, `uv add pydantic-monty`), `@pydantic/monty` (npm), and the `monty-pool` Rust crate (`cargo add monty-pool`) providing a Monty worker pool and session.feed_run()/feedRun() to execute LLM-generated Python with no filesystem, environment, or network access. Supports `inputs` and `external_lookup` / `externalLookup` to pass host-side values and callable tools into the sandbox, and `feed_start` + dump() + `load_snapshot` to suspend, serialize (including the call stack), and resume a paused interpreter on another machine. Enforces VM-level resource limits via `max_memory`, `max_duration_secs`, `max_recursion_depth`, and pool-level `max_suspensions`. A commercial Full Monty option exposes the same workers behind a WebSocket as a container image for OS-level isolation and horizontal scaling.

Run LLM-generated Python that calls a host-side tool (e.g. a nutrition lookup) without giving the sandbox any filesystem or network access.

```python
from pydantic_monty import Monty

code = """
kcal = nutrition('chocolate bar')['kcal']
hours = kcal * 4184 / (bulb_watts * 3600)
print(f'a chocolate bar could power a {bulb_watts}W bulb for
{hours:.1f} hours')
"""

with Monty() as pool:
    with pool.checkout() as session:
        session.feed_run(
            code,
            inputs={'bulb_watts': 10},
            external_lookup={'nutrition': lambda food: {'kcal':
230}},
        )
```

Serialise a paused sandbox to bytes after a host call so the interpreter state can be stored and resumed later on a different machine.

```python
from pydantic_monty import Monty

with Monty() as pool:
    with pool.checkout() as session:
        suspension = session.feed_start(code, inputs={'x': 42})
        snapshot_bytes = suspension.dump()

# Later, on any machine:
    with pool.checkout() as session2:
        session2.load_snapshot(snapshot_bytes)
```

— Full mechanism, limits, and dual-language APIs all named with code `product docs`

02 **Declarative Agent Specs via Agent.from_file/from_spec** NEW

98

Adds Agent.from_file('agent.yaml') and Agent.from_spec(dict_or_AgentSpec, **kwargs) to build agents from YAML/JSON without Python construction code, with kwargs overriding spec fields (`model` , `instructions` , `capabilities` , `model_settings` , `output_type` , `deps_type` , `retries`). Adds `AgentSpec.from_file` for finer-grained loading and AgentSpec.to_file('agent.yaml') to serialize a spec plus an optional companion JSON Schema (`schema_path=None` to skip). Adds `TemplateStr` , a

Handlebars-style (`{{variable}}`) template type for `instructions` / `description` rendered against `deps` at runtime — any `{{` -containing string in a spec file is auto-promoted. The `AgentSpec` model carries `model` , `name` , `description` , `instructions` , `model_settings` , `capabilities` , `deps_schema` , `output_schema` , `retries` (`int` or `AgentRetries`), `end_strategy` (`'early'` / `'graceful'` / `'exhaustive'`), `tool_timeout` , `instrument` , and `metadata` ; `output_schema` yields a `StructuredDict` returning `dict[str, Any]` when no `output_type` is passed, and `deps_schema` validates `TemplateStr` variable names without a Python `deps_type` . Declarative capability syntax supports `Thinking` , `Instrumentation` , `WebSearch` , `WebFetch` , `ImageGeneration` , `XSearch` , `MCP` , `ToolSearch` , `PrefixTools` , `NativeTool` , `IncludeToolReturnSchemas` , `SetToolMetadata` , `RaiseContentFilterError` , and `ReinjectSystemPrompt` .

Define a research agent in a YAML file so prompt engineers can tune model, instructions, and capabilities without touching Python code.

```yaml
model: anthropic:claude-opus-4-6
instructions: You are a helpful research assistant.
model_settings:
  max_tokens: 8192
capabilities:
  - WebSearch:
      local: duckduckgo
  - Thinking:
      effort: high
```

Load a YAML spec at startup and run it — zero agent construction code in your application.

```python
from pydantic_ai import Agent

agent = Agent.from_file('agent.yaml')
result = agent.run_sync('Summarize the latest news on LLM
safety.')
print(result.output)
```

Override spec fields at load time to inject a typed deps context and personalise instructions via `TemplateStr` — useful when the same spec file powers multiple tenant-specific deployments.

```python
from dataclasses import dataclass
from pydantic_ai import Agent

@dataclass
class UserContext:
    user_name: str

agent = Agent.from_spec(
    {
        'model': 'anthropic:claude-opus-4-6',
        'instructions': 'You are helping {{user_name}}.',
        'capabilities': [{'WebSearch': {'local':
'duckduckgo'}}],
    },
    deps_type=UserContext,
)
result = agent.run_sync('Find recent papers on RAG.',
deps=UserContext(user_name='Alice'))
print(result.output)
```

— Every spec field and loader function named with runnable examples `product docs`

03 **Hooks capability for intercepting model/tool/run lifecycle** NEW

98

Adds Hooks (`pydantic_ai.capabilities`) with `@hooks.on.*` decorator registration (or constructor kwargs, e.g. Hooks(before_model_request=log_request)) — no subclassing required. Provides hook families for model requests (`before_model_request` , `after_model_request` , `model_request` , `model_request_error` , with `ModelRequestContext` bundling `model` , `messages` , `model_settings` , `model_request_parameters`), tool validation (`before_tool_validate` , `after_tool_validate` , `tool_validate` , `tool_validate_error`), tool execution (`before_tool_execute` , `after_tool_execute` , `tool_execute` , `tool_execute_error`), run-level (`before_run` , `after_run` , `run` , `run_error`), node-level (`before_node_run` , `after_node_run` , `node_run` , `node_run_error` for `UserPromptNode` / `ModelRequestNode` / `CallToolsNode`), output validation (`before_output_validate` , `after_output_validate` , `output_validate` , `output_validate_error`), and output processing (`before_output_process` , `after_output_process` , `output_process` , `output_process_error`). Also adds `prepare_tools` / `prepare_output_tools` to filter tool definitions per step, `deferred_tool_calls` to resolve approval-required tools inline via `DeferredToolRequests` / `DeferredToolResults` , and `run_event_stream` / `event` to wrap or observe the event stream (`ToolCallPart` , `EnqueuedMessagesEvent` ,

`CustomEvent`, `RealtimeEvent`). Supports `defer_loading=True` for on-demand hooks, and raising `SkipModelRequest`, `SkipToolValidation`, or `SkipToolExecution` to bypass the corresponding step; hooks may be sync (run in a thread pool) or async.

Log every outbound LLM request with message count — useful for debugging prompt sizes in production.

```python
from pydantic_ai import Agent, ModelRequestContext, RunContext
from pydantic_ai.capabilities import Hooks

hooks = Hooks()

@hooks.on.before_model_request
async def log_request(ctx: RunContext, request_context:
ModelRequestContext) -> ModelRequestContext:
    print(f'Sending {len(request_context.messages)} messages to
the model')
    return request_context

agent = Agent('openai:gpt-4o', capabilities=[hooks])
result = agent.run_sync('Summarize this report.')
print(result.output)
```

Auto-approve all deferred tool calls inline during a run — useful for CI pipelines where human approval is not available.

python

```python
from pydantic_ai import Agent, DeferredToolRequests,
DeferredToolResults, RunContext
from pydantic_ai.capabilities import Hooks

hooks = Hooks()

@hooks.on.deferred_tool_calls
async def auto_approve(ctx: RunContext, *, requests:
DeferredToolRequests) -> DeferredToolResults:
    return requests.build_results(approve_all=True)

agent = Agent('openai:gpt-4o', capabilities=[hooks])

@agent.tool_plain(requires_approval=True)
def delete_file(path: str) -> str:
    return f'File {path!r} deleted'

result = agent.run_sync('Delete tmp/cache.log')
print(result.output)
```

Count only `PartStartEvent` occurrences in the event stream to measure how many response parts a model emits per run.

python

```python
from pydantic_ai import Agent, PartStartEvent, RunContext
from pydantic_ai.capabilities import Hooks

hooks = Hooks()
event_count = 0

@hooks.on.event(PartStartEvent)
async def count_events(ctx: RunContext, event: PartStartEvent) -
> None:
    global event_count
    event_count += 1

agent = Agent('openai:gpt-4o', capabilities=[hooks])
agent.run_sync('Tell me three facts about the ocean.')
print(f'PartStartEvent count: {event_count}')
```

— Every hook name and bypass exception enumerated with runnable code product docs

04 **Custom capabilities via AbstractCapability** NEW 96

Adds `AbstractCapability` base class with override points `get_toolset`, `get_native_tools`, `get_wrapper_toolset`, `get_instructions`, and `get_model_settings` for building custom agent capabilities — `get_wrapper_toolset` wraps the whole assembled toolset via `WrapperToolset` for cross-cutting behaviors like logging, `get_toolset` accepts a prebuilt `AbstractToolset` or a `RunContext` callable, and `get_instructions` / `get_model_settings` support static values, `TemplateStr`, or per-run callables. Supports `defer_loading=True` (requiring a stable `id`); capability `id` drives merge semantics — same-`id` instances merge field-by-field, and a run-level capability overrides an agent-level one outright unless a custom `combine` override point is defined. Instruction parts are keyed `'capability:<id>'` or `'capability:<id>:<name>'` via @capability.instructions(name=...). Supports `AbstractCapability[MyDeps]` generic typing for typed `RunContext[MyDeps]`, and plain class, `@dataclass`, or custom `__init__` construction patterns.

Intercept every tool call across an agent to add audit logging without modifying individual tools.

python

```python
from dataclasses import dataclass
from typing import Any
from pydantic_ai import Agent
from pydantic_ai.capabilities import AbstractCapability
from pydantic_ai.toolsets import AbstractToolset
from pydantic_ai.toolsets.wrapper import WrapperToolset

@dataclass
class LoggingToolset(WrapperToolset[Any]):
    async def call_tool(self, tool_name: str, tool_args:
dict[str, Any], *args: Any, **kwargs: Any) -> Any:
        print(f'[AUDIT] Calling tool: {tool_name} args=
{tool_args}')
        return await super().call_tool(tool_name, tool_args,
*args, **kwargs)

@dataclass
class AuditLogCalls(AbstractCapability[Any]):
    def get_wrapper_toolset(self, toolset: AbstractToolset[Any])
-> AbstractToolset[Any]:
        return LoggingToolset(wrapped=toolset)

agent = Agent('openai:gpt-5.2', capabilities=[AuditLogCalls()])
```

Inject dynamic per-run instructions (e.g., current timestamp) into an agent without hardcoding a static string.

```python
from dataclasses import dataclass
from datetime import datetime
from typing import Any
from pydantic_ai import Agent, RunContext
from pydantic_ai.capabilities import AbstractCapability

@dataclass
class CurrentTimeContext(AbstractCapability[Any]):
    def get_instructions(self):
        def _instructions(ctx: RunContext[Any]) -> str:
            return f'The current UTC time is
{datetime.utcnow().isoformat()}. Use it when answering time-
sensitive questions.'
        return _instructions

agent = Agent('openai:gpt-5.2', capabilities=
[CurrentTimeContext()])
result = agent.run_sync('Is the market open right now?')
```

Bundle a reusable set of tools into a packaged capability so any agent can adopt them with a single capabilities= entry.

```python
from dataclasses import dataclass
from typing import Any
from pydantic_ai import Agent
from pydantic_ai.capabilities import AbstractCapability
from pydantic_ai.toolsets import FunctionToolset, AgentToolset

math_toolset = FunctionToolset()

@math_toolset.tool_plain
def add(a: float, b: float) -> float:
    """Add two numbers."""
    return a + b

@math_toolset.tool_plain
def multiply(a: float, b: float) -> float:
    """Multiply two numbers."""
    return a * b

@dataclass
class MathTools(AbstractCapability[Any]):
    id: str | None = 'math_tools'
    def get_toolset(self) -> AgentToolset[Any] | None:
        return math_toolset

agent = Agent('openai:gpt-5.2', capabilities=[MathTools()])
result = agent.run_sync('What is 7 multiplied by 6?')
```

— All override points and merge semantics named with code `product docs`

05

xAI/Grok integration with X Search, image generation, and reasoning controls

NEW

96

Adds `XaiModel` / `XaiProvider` (`pydantic_ai.models.xai` / `pydantic_ai.providers.xai`) for `'xai:<model-name>'` agents authenticated via `XAI_API_KEY` , with provider options `api_host` , `timeout` , and `metadata` (e.g. `x-grok-conv-id` for prompt-cache sticky routing) and a custom `xai_sdk.AsyncClient` via `xai_client` . Adds the `XSearch` capability (`allowed_x_handles` , `excluded_x_handles` , `from_date` , `to_date` , `enable_image_understanding` , `enable_video_understanding` , `include_output`) for real-time X search, plus `XaiModelSettings` fields `xai_include_x_search_output` , `xai_reasoning_effort` (`'none'|'low'|'medium'|'high'` , for Grok 4.3), `xai_max_turns` (caps server-side tool-loop turns), `xai_agent_count` (parallel agents for `grok-4.20-multi-agent`), and `xai_include_attachment_search_output` (default `False` , surfaces the

`attachment_search` / `pdf_browse` tool output). Adds `ImageGenerator` support for xAI image models via `XaiImageGenerationSettings` (`aspect_ratio`, `xai_resolution`) with moderation-aware batching that reports flagged images in `provider_details['moderated_image_indices']` and only raises `ContentFilterError` when every image is flagged. Installs via `pydantic-ai-slim[xai]`.

Run a Grok agent that searches X for recent posts from specific AI company accounts, with image understanding and programmatic access to the raw search results.

```python
from datetime import datetime
from pydantic_ai import Agent
from pydantic_ai.capabilities import XSearch

agent = Agent(
    'xai:grok-4.3',
    capabilities=[
        XSearch(
            allowed_x_handles=['OpenAI', 'AnthropicAI'],
            from_date=datetime(2025, 1, 1),
            enable_image_understanding=True,
            include_output=True,
        )
    ],
)
result = agent.run_sync('Summarize recent AI announcements.')
print(result.output)
```

Cap xAI server-side agentic tool turns and set reasoning depth to keep costs predictable on a Grok 4.3 agent doing web research.

```python
from pydantic_ai import Agent
from pydantic_ai.models.xai import XaiModelSettings

agent = Agent(
    'xai:grok-4.3',
    model_settings=XaiModelSettings(
        xai_reasoning_effort='low',
        xai_max_turns=3,
    ),
)
result = agent.run_sync('What are the top cybersecurity threats
this week?')
print(result.output)
```

Pin a multi-turn conversation to a single xAI prompt-cache node via `x-grok-conv-id` to avoid reprocessing repeated prefixes.

```python
from pydantic_ai import Agent
from pydantic_ai.models.xai import XaiModel
from pydantic_ai.providers.xai import XaiProvider

provider = XaiProvider(
    api_key='your-api-key',
    metadata=(('x-grok-conv-id', 'session-abc123'),),
)
model = XaiModel('grok-4.3', provider=provider)
agent = Agent(model)
result = agent.run_sync('Continue our analysis of the incident report.')
print(result.output)
```

— Extensive named settings and three runnable code examples `product docs`

06 **Transport-layer retry framework via pydantic_ai.retries** NEW

96

Adds `AsyncHTTPX2TenacityTransport` and `HTTPX2TenacityTransport` (`pydantic_ai.retries`) to attach tenacity-powered retries directly to `httpx2` clients used by any provider, configured via a unified `RetryConfig` object wrapping tenacity's `retry` , `wait` , `stop` , and `reraise` parameters. Adds `wait_retry_after` , a wait strategy parsing provider Retry-After headers (seconds or HTTP-date format) from 429s with a configurable `max_wait` ceiling and fallback to any tenacity wait strategy, plus `ModelHTTPError.retry_after` for custom backoff outside a transport. Installs via `pydantic-ai-slim[retries]` (pulls in `tenacity`). Documentation now enumerates seven retry layers — transport, provider SDK, durable execution (`retry_policy` in Temporal's `ActivityConfig` , `max_attempts` in DBOS's `StepConfig` , `retries` in Prefect's `TaskConfig`), model fallback (`FallbackModel`), tool retries (`retries={'tools': N}`), output retries (`retries={'output': N}` /ToolOutput(max_retries=N)), and `ModelRetry` `-raising` hooks — clarifying only the last three consume model round trips and that `UsageLimits.request_limit` counts only model requests.

Attach rate-limit-aware retries to an OpenAI provider so 429s are automatically retried up to 5 times, honoring the provider's Retry-After header before falling back to exponential backoff.

```python
from httpx2 import AsyncClient, HTTPStatusError
from tenacity import retry_if_exception_type,
stop_after_attempt, wait_exponential
from pydantic_ai import Agent
from pydantic_ai.models.openai import OpenAIChatModel
from pydantic_ai.providers.openai import OpenAIProvider
from pydantic_ai.retries import AsyncHTTPX2TenacityTransport,
RetryConfig, wait_retry_after

transport = AsyncHTTPX2TenacityTransport(
    config=RetryConfig(
        retry=retry_if_exception_type(HTTPStatusError),
        wait=wait_retry_after(
            fallback_strategy=wait_exponential(multiplier=1,
max=60),
            max_wait=300,
        ),
        stop=stop_after_attempt(5),
        reraise=True,
    ),
    validate_response=lambda r: r.raise_for_status(),
)
client = AsyncClient(transport=transport)
model = OpenAIChatModel('gpt-4o',
provider=OpenAIProvider(http_client=client))
agent = Agent(model)
```

Retry only on transient network errors (timeouts, connection resets, read failures) with exponential backoff, leaving 4xx client errors to surface immediately.

```python
import httpx2
from tenacity import retry_if_exception_type,
stop_after_attempt, wait_exponential
from pydantic_ai.retries import AsyncHTTPX2TenacityTransport,
RetryConfig

transport = AsyncHTTPX2TenacityTransport(
    config=RetryConfig(
        retry=retry_if_exception_type((
            httpx2.TimeoutException,
            httpx2.ConnectError,
            httpx2.ReadError,
        )),
        wait=wait_exponential(multiplier=1, max=10),
        stop=stop_after_attempt(3),
        reraise=True,
    )
)
client = httpx2.AsyncClient(transport=transport)
```

— Names classes, config keys and header behaviour with runnable code `product docs`

---

07 **Agent-free model requests via pydantic_ai.direct** NEW

86

Adds `model_request_sync`, `model_request`, `model_request_stream`, and `model_request_stream_sync` in `pydantic_ai.direct` for making sync/async, streamed/non-streamed LLM requests without constructing an `Agent`, supporting `ModelRequestParameters` (`function_tools`, `allow_text_output`). Supports per-call OpenTelemetry/Logfire instrumentation via `instrument=True`.

Make a synchronous LLM request without spinning up an Agent — useful for lightweight scripts or custom abstractions.

```python
from pydantic_ai import ModelRequest
from pydantic_ai.direct import model_request_sync

model_response = model_request_sync(
    'anthropic:claude-haiku-4-5',
    [ModelRequest.user_text_prompt('What is the capital of
France?')]
)
print(model_response.parts[0].content)
print(model_response.usage)
```

Call an LLM with tool definitions (Pydantic-generated JSON schema) for function calling without an Agent.

```python
from typing import Literal
from pydantic import BaseModel
from pydantic_ai import ModelRequest, ToolDefinition
from pydantic_ai.direct import model_request
from pydantic_ai.models import ModelRequestParameters

class Divide(BaseModel):
    """Divide two numbers."""
    numerator: float
    denominator: float
    on_inf: Literal['error', 'infinity'] = 'infinity'

model_response = await model_request(
    'openai:gpt-5-nano',
    [ModelRequest.user_text_prompt('What is 123 / 456?')],
    model_request_parameters=ModelRequestParameters(
        function_tools=[
            ToolDefinition(
                name=Divide.__name__.lower(),
                description=Divide.__doc__,

parameters_json_schema=Divide.model_json_schema(),
            )
        ],
        allow_text_output=True,
    ),
)
```

Enable OpenTelemetry tracing on a single direct model call without global instrumentation.

```python
import logfire
from pydantic_ai import ModelRequest
from pydantic_ai.direct import model_request_sync

logfire.configure()

model_response = model_request_sync(
    'anthropic:claude-haiku-4-5',
    [ModelRequest.user_text_prompt('What is the capital of
France?')],
    instrument=True
)
print(model_response.parts[0].content)
```

— All four functions and their parameters named with code `product docs`

---

08 **Hugging Face Inference Providers support** NEW 83

Adds `HuggingFaceModel` (`pydantic_ai.models.huggingface`) and `HuggingFaceProvider` (`pydantic_ai.providers.huggingface`) to run open-source models such as `Qwen/Qwen3-235B-A22B` and DeepSeek R1 through Hugging Face Inference Providers, selecting a backend (Cerebras, Together AI, Cohere, Nebius, Fireworks AI, etc.) via `provider_name` or a custom `AsyncInferenceClient` passed as `hf_client` (exposing `headers` , `bill_to` , `base_url`). Reads `HF_TOKEN` for auth and installs via the `pydantic-ai-slim[huggingface]` extras group.

Pin inference to a specific backend (e.g. Nebius) and supply credentials in code rather than via environment variables.

```python
from pydantic_ai import Agent
from pydantic_ai.models.huggingface import HuggingFaceModel
from pydantic_ai.providers.huggingface import
HuggingFaceProvider

model = HuggingFaceModel(
    'Qwen/Qwen3-235B-A22B',
    provider=HuggingFaceProvider(api_key='hf_token',
provider_name='nebius'),
)
agent = Agent(model)
```

Bill inference costs to an HF organization and route through a specific provider using a custom AsyncInferenceClient.

python

```python
from huggingface_hub import AsyncInferenceClient
from pydantic_ai import Agent
from pydantic_ai.models.huggingface import HuggingFaceModel
from pydantic_ai.providers.huggingface import
HuggingFaceProvider

client = AsyncInferenceClient(
    bill_to='openai',
    api_key='hf_token',
    provider='fireworks-ai',
)
model = HuggingFaceModel(
    'Qwen/Qwen3-235B-A22B',
    provider=HuggingFaceProvider(hf_client=client),
)
agent = Agent(model)
```

— Named classes and backend routing with two code examples product docs

09 **Mistral model provider support** NEW 83

Adds `MistralModel` (`pydantic_ai.models.mistral`) and `MistralProvider` (`pydantic_ai.providers.mistral`) to run agents against Mistral-hosted models like `mistral-large-latest` and `mistral-small-latest`, authenticated via `MISTRAL_API_KEY` or explicit `api_key` / `base_url` / `http_client` arguments. Installable via the `mistral` extras group (`pip install 'pydantic-ai-slim[mistral]'`), exposes `LatestMistralModelNames` for reference, and accepts an `httpx2.AsyncClient` as `http_client` (legacy `httpx.AsyncClient` still works but is deprecated for removal in v3).

Run a PydanticAI agent against Mistral using the shorthand model-name string and an environment variable for auth.

python

```python
from pydantic_ai import Agent
agent = Agent('mistral:mistral-large-latest')
```

Route requests to a self-hosted or private Mistral endpoint with a custom base URL and a 30-second HTTP timeout.

python

```python
from httpx2 import AsyncClient
from pydantic_ai import Agent
from pydantic_ai.models.mistral import MistralModel
from pydantic_ai.providers.mistral import MistralProvider

custom_http_client = AsyncClient(timeout=30)
model = MistralModel(
    'mistral-large-latest',
    provider=MistralProvider(
        api_key='your-api-key',
        base_url='https://<mistral-provider-endpoint>',
        http_client=custom_http_client,
    ),
)
agent = Agent(model)
```

— Classes, env var, extras and deprecation path all named `product docs`

---

10 **Groq model provider support** NEW 83

Adds `GroqModel` and `GroqProvider` (`pydantic_ai.models.groq` / `pydantic_ai.providers.groq`) for Groq-hosted LLMs, configured via `GROQ_API_KEY` or an `api_key` argument, with `'groq:<model-name>'` shorthand. Supports a custom `httpx.AsyncClient` via `http_client` for timeouts/transport, and exposes the Groq SDK's retry count through groq_client=AsyncGroq(max_retries=...) (default `max_retries=2`). Installs via `pydantic-ai-slim[groq]` .

Use a custom HTTP client with a strict timeout when Groq requests must complete within a pipeline SLA.

python

```python
from httpx import AsyncClient
from pydantic_ai import Agent
from pydantic_ai.models.groq import GroqModel
from pydantic_ai.providers.groq import GroqProvider

custom_http_client = AsyncClient(timeout=30)
model = GroqModel(
    'llama-3.3-70b-versatile',
    provider=GroqProvider(api_key='your-api-key',
http_client=custom_http_client),
)
agent = Agent(model)
```

Disable the Groq SDK's built-in retries when your own transport layer already handles retry logic.

```python
from groq import AsyncGroq
from pydantic_ai import Agent
from pydantic_ai.models.groq import GroqModel
from pydantic_ai.providers.groq import GroqProvider

model = GroqModel(
    'llama-3.3-70b-versatile',
    provider=GroqProvider(groq_client=AsyncGroq(max_retries=0)),
)
agent = Agent(model)
```

— Retry default and transport override both named with code `product docs`

11 **Cohere model provider support** NEW 82

Adds `CohereModel` (`pydantic_ai.models.cohere`), `CohereProvider` (`api_key`, `http_client`), and `CohereModelSettings` (`temperature`, `top_k`) for running agents against Cohere endpoints via `CO_API_KEY` or the `cohere:` shorthand prefix. Installs via `pydantic-ai-slim[cohere]`; Cohere's built-in client retries server errors and rate limits twice regardless of transport, and `max_retries` cannot be configured.

Use a custom HTTP client with a longer timeout to avoid transport-layer failures on slow Cohere responses.

```python
from httpx import AsyncClient
from pydantic_ai import Agent
from pydantic_ai.models.cohere import CohereModel
from pydantic_ai.providers.cohere import CohereProvider

custom_http_client = AsyncClient(timeout=30)
model = CohereModel(
    'command-r7b-12-2024',
    provider=CohereProvider(api_key='your-api-key',
http_client=custom_http_client),
)
agent = Agent(model)
```

Tune sampling behaviour for a Cohere agent using `CohereModelSettings` to get more deterministic outputs.

```python
from pydantic_ai import Agent
from pydantic_ai.models.cohere import CohereModel,
CohereModelSettings

model = CohereModel('command-r7b-12-2024')
settings = CohereModelSettings(temperature=0.2, top_k=40)
agent = Agent(model, model_settings=settings)
```

— Settings class and fixed-retry caveat both named `product docs`

12 **PydanticAI skill for coding agents** NEW 81

Ships an installable skill giving coding agents up-to-date PydanticAI knowledge, covering tools, capabilities, structured output, streaming, testing, multi-agent delegation, hooks, and agent specs. Installable via `claude plugin install pydantic-ai@claude-plugins-official`, or `claude plugin marketplace add pydantic/skills` + `claude plugin install ai@pydantic-skills`, via the agentskills.io standard with `npx skills add pydantic/skills` (30+ agents including Claude Code, Codex, Cursor, Gemini CLI), or from a project's transitive dependency via `uvx library-skills --all` (add `--claude` to also write into `.claude/skills/`).

Install the PydanticAI skill for all compatible agents in a project from its bundled transitive dependency, also writing into Claude Code's expected directory.

```
$ uvx library-skills --all --claude
```

Give a Claude Code agent up-to-date PydanticAI knowledge via the official Anthropic marketplace plugin.

```
$ claude plugin install pydantic-ai@claude-plugins-official
```

Install the PydanticAI skill for Cursor, Codex, Gemini CLI, or any other agentskills.io-compatible agent in your project.

```
$ npx skills add pydantic/skills
```

— Four exact install commands given, thin on internal mechanism `product docs`

13 **Standalone image generation via ImageGenerator** NEW 65

Adds an `ImageGenerator` class with a generate_sync() method to generate images directly without an agent run, using providers such as `openai:gpt-image-2` and returning a result with `.image.data` bytes.

Generate a standalone image from a text prompt and save it to disk without spinning up an agent run.

```python
from pydantic_ai import ImageGenerator
from pathlib import Path

generator = ImageGenerator('openai:gpt-image-2')
result = generator.generate_sync('A minimalist logo for a coffee shop called Extract.')
Path('logo.png').write_bytes(result.image.data)
```

Generate a standalone image from a prompt without spinning up a full agent run.

```python
from pydantic_ai import ImageGenerator
from pathlib import Path

generator = ImageGenerator('openai:gpt-image-2')
result = generator.generate_sync('A minimalist logo for a coffee shop called Extract.')
Path('logo.png').write_bytes(result.image.data)
```

— One method and one example provider given, no size/format options v2.41.0

THINNER COVERAGE BELOW

14 **Cerebras model provider support** NEW 55

Adds `CerebrasModel` (`pydantic_ai.models.cerebras`) and `CerebrasProvider` (`pydantic_ai.providers.cerebras`) for Cerebras API integration, configured via `CEREBRAS_API_KEY` or an explicit `provider` argument, installable via `pydantic-ai-slim[cerebras]` .

Run an agent against the Cerebras API using llama-3.3-70b with a custom HTTP timeout.

```python
from httpx2 import AsyncClient
from pydantic_ai import Agent
from pydantic_ai.models.cerebras import CerebrasModel
from pydantic_ai.providers.cerebras import CerebrasProvider

custom_http_client = AsyncClient(timeout=30)
model = CerebrasModel(
    'llama-3.3-70b',
    provider=CerebrasProvider(api_key='your-api-key',
http_client=custom_http_client),
)
agent = Agent(model)
```

— Only class names and env var, no extra settings described `product docs`

---

15 **openai-codex provider for ChatGPT/Codex subscription auth** NEW

38

Adds an `openai-codex` provider that authenticates via ChatGPT/Codex subscriptions instead of API keys.

— No code example or configuration detail given `v2.41.0`

---

16 **fallback_model deprecated on ImageGeneration and XSearch**

DEPRECATED

30

Deprecates `fallback_model` in favor of `fallback_subagent_model` on `ImageGeneration` and `XSearch`.

— Names old/new field only, no migration guidance `v2.41.0`

WAS THIS USEFUL? 👍 👎

Nous Research Hermes

Feedback ⓘ 1 RELEASE · 2026-09-07

NOTES ↗ CODE ↗ RANK

A terminal-native, self-improving AI agent for autonomous coding and tasks, with persistent memory, agent-created skills, and a multi-platform messaging gateway.

> Hermes shipped a large batch of terminal UX and reliability upgrades this window: a full-screen subagent monitor with mid-turn message steering, a configurable shared directory for agent-created skills, deferred background reviews for local GPU setups, and retained output for background commands. The messaging gateway also gained PII redaction, authoritative platform disabling, flood-control auto-retry, and a proxy trust toggle, alongside new asynchronous peer-run commands, Fast Mode acceleration for OpenAI/xAI/Anthropic, and several new provider/model routing targets.

└──▷ WHAT SHIPPED · 35 FEATURES most completely described first ? what's the number?

01 Retained results for completed background commands IMPROVED

90

Retains exit status and captured output of completed background commands, accessible via process(action="log") for output and process(action="poll") for exit status from the conversation that launched it; process(action="list") now includes retained results, not just live processes. Keeps the newest 64 completed results for up to 7 days under `logs/process-results/` in the profile's Hermes home, with each receipt capped at a 200,000-character output tail and terminal secret-redaction always applied.

Retrieve output from a background command that finished while you were away — useful when a long-running job completed after the parent conversation was compressed or resumed.

```
$ process(action="log")
```

— Exact actions, limits, path, and example command given product docs

02 Asynchronous peer run, status, and stop commands NEW

90

Adds `hermes peer run <peer>[/<agent>] [message]` to start a long Bot Chat turn asynchronously and return its `run_id`, session ID, and idempotency key; `hermes peer status <peer>[/<agent>] <run_id>` polls a run and prints its final output when complete; `hermes peer stop <peer>[/<agent>] <run_id>` stops an exact run without targeting another turn.

Kick off a long-running agent task from a script and poll for its result without blocking the caller.

```
$ run_id=$(hermes peer run myagent/researcher 'Summarize recent CVEs in openssl' | jq -r .run_id)
hermes peer status myagent/researcher "$run_id"
```

— Three exact subcommands with a runnable script example `product docs`

---

03 **Configurable shared directory for agent-created skills** NEW

85

Adds `skills.create_dir` config key to redirect new agent-created skills to a custom directory — such as a shared `brain` directory, a git-tracked repo, or a fleet-wide skills volume — instead of the default profile-local skills directory. Agent-facing tool descriptions and prompt text dynamically render the configured `create_dir` path so the agent is automatically told to write skills there, and skills under `create_dir` are fully integrated into the skill index, slash commands, and support patch and delete operations alongside local skills.

Route all agent-created skills to a git-tracked shared directory so teammates and fleet agents share the same skill library.

```yaml
create_dir: /opt/brain/skills
```

— Named config key with full integration mechanism and example `product docs`

---

04 **Deferred background reviews for local GPU models** NEW

85

Adds `defer` config option (`auto` or `never`) for `auxiliary.background_review` so reviews on the managed local `llama-server` queue at turn end and run once the machine is idle, preventing GPU contention with the next prompt; `defer_max_age_s` forces a queued review to run after a maximum wait even if the machine never goes idle.

Prevent local GPU contention by deferring background reviews until the machine is idle, with a safety valve to force execution after 10 minutes.

```yaml
defer: auto
defer_max_age_s: 600
```

Disable automatic deferred reviews entirely on a local model setup so no background GPU work runs without explicit trigger.

```yaml
defer: never
```

— Named config keys with mechanism and two worked examples product docs

05 **Authoritative platform disable via config** BREAKING 85

Makes `platforms.<name>.enabled: false` in `config.yaml` authoritative over credentials present in `.env` for twelve platforms (Weixin, WhatsApp Cloud, Home Assistant, Email, SMS, DingTalk, Feishu, WeCom, WeCom callback, BlueBubbles, QQ Bot, Yuanbao); previously credentials alone re-enabled the adapter regardless of this key. Startup now logs a WARNING per affected platform instead of silently re-enabling it.

Explicitly disable the Weixin adapter while keeping its credentials in .env available for send-only tooling.

```yaml
platforms:
  weixin:
    enabled: false
```

Explicitly disable a platform adapter even though its token remains in the environment, to prevent it from starting while keeping send-only tooling functional.

```yaml
platforms:
  weixin:
    enabled: false
```

— Lists all affected platforms with exact config key and example product docs

06 **PII redaction for gateway message context** NEW 80

Adds `privacy.redact_pii: true` config key to hash platform, chat, thread, sender, message, profile, and scope identifiers in the model-visible per-message JSON context on supported platforms, while preserving original identifiers internally for routing.

Hash sender, chat, and thread identifiers visible to the model on supported platforms to limit PII exposure in agent context.

```yaml
privacy:
  redact_pii: true
```

Hash sender and chat identifiers visible to the model when handling messages from privacy-sensitive platforms.

```yaml
privacy:
  redact_pii: true
```

— Named config key, exact fields hashed, and example product docs

07 **Proxy trust control for gateway adapters** NEW 80

Adds `gateway.trust_env` config key to disable inherited `HTTP_PROXY` / `HTTPS_PROXY` / `NO_PROXY` / `SSL_CERT_FILE` and macOS system proxy auto-detection for all platform adapters at once, while explicit per-platform proxy variables (`DISCORD_PROXY` , `TELEGRAM_PROXY` , `MATRIX_PROXY` , ...) remain honored.

Prevent a gateway running under a service manager from failing to connect through a local proxy listener that may not be running.

```yaml
gateway:
  trust_env: false
```

Prevent a gateway running as a Windows Scheduled Task from inheriting a proxy (e.g. a local Clash listener) that isn't available in that context.

```yaml
gateway:
  trust_env: false
```

— Named config key and env vars with worked example product docs

08 **Fast Mode across OpenAI, xAI, and Anthropic** NEW 80

Adds Fast Mode support for OpenAI Priority Processing (`service_tier: priority`), xAI Priority Processing on Grok 4.6, and Anthropic Fast Mode (`speed: fast` , Opus 4.8 / Opus 5 only), configurable with `normal` , `fast` , `auto` , `cold` modes and a `fast_auto_seconds` window. The `/fast normal|fast|auto|cold` slash command switches Fast Mode for the current session, with an optional flag to persist the setting; `/fast` alone shows the current mode.

— Names providers, fields, and slash command syntax product docs

09 **Fleet health check via runs incidents** NEW 80

Adds `hermes runs incidents` command as a read-only fleet health check for failed runs, failed deliveries, overdue/missing `next_run_at` , and missing scripts or workdirs

— it exits non-zero when issues are found.

Check fleet health in CI — exits non-zero if any runs have failed or scheduled tasks are overdue.

```
$ hermes runs incidents
```

— Exact command with CI-usable exit-code behavior and example `product docs`

---

10 **Timeout for slow context file reads** NEW 75

Adds `context_file_read_timeout` config key (default 5 seconds) to skip context files that take too long to read — such as those on iCloud Drive, OneDrive, or NFS — with a warning instead of blocking, bounding per-file reads on network-backed filesystems before the system prompt is built.

Raise the read timeout when scanning context files on a slow NFS or cloud-synced drive so they are not silently skipped during analysis.

```yaml
context_file_read_timeout: 15
```

— Named config key, default value, and example provided `product docs`

---

11 **Docker snap compatibility for sandbox flags** NEW 75

Adds `docker_snap_compat` config key (also exposed as `TERMINAL_DOCKER_SNAP_COMPAT` env var) to drop `--init` and `--no-new-privileges` sandbox flags on hosts where Docker is installed as a snap (e.g. Ubuntu Azure VMs), preventing container startup failures.

Enable sandbox compatibility on a snap-packaged Docker host (e.g. an Ubuntu Azure VM) so Hermes containers start successfully.

```yaml
docker_snap_compat: true
```

— Named config key, env var, and example value `product docs`

---

12 **Loop-detection hard stops for unattended sessions** NEW 75

Adds `non_interactive_hard_stop_enabled` config key to control loop-detection hard stops, enabling them by default for unattended gateway and cron sessions while leaving interactive CLI, TUI, Desktop, and ACP sessions in warning-only mode.

Opt an unattended gateway deployment out of hard stops if your workflow involves legitimate retries that would otherwise be blocked.

```yaml
non_interactive_hard_stop_enabled: false
```

— Named config key with default scope and example product docs

13 **Backend port-conflict detection and ephemeral binding** NEW

75

Adds port-conflict detection to the backend: prints a machine-readable `BACKEND_PORT_IN_USE port=<port>` sentinel and exits with `EX_TEMPFAIL` when the requested port is occupied; supports `--port 0` to bind a free ephemeral port, announced via `HERMES_BACKEND_READY port=<port>`.

— Exact sentinels and flag named, no worked example product docs

14 **Full-screen subagent monitor and dock toggle** NEW 70

Adds `F6` to open a full-screen live subagent monitor without losing the composer draft, with arrow-key worker selection, a log view, `s` to steer a worker, and `x` to stop one with confirmation; `F7` toggles the live subagent dock between a multi-row preview and a single summary line without moving composer focus.

— Named keybinds and controls but no worked example product docs

15 **Mid-turn message steering with background handoff** NEW

65

Sending a message while a turn is in progress redirects it: model generation restarts with reasoning and completed work preserved, and any running foreground terminal command is moved to the background instead of being killed, with a completion notification when it finishes.

Redirect an in-progress agent turn without waiting for a running build or long-lived command to finish — the command is handed to the background and the agent reads your message immediately.

> 📍 While the agent is running a foreground command (e.g. a build or poller), type a new message in the composer and send it. The command moves to the background; you receive a completion notification when it finishes, and the agent acts on your message right away.

— Clear mechanism and worked example but no named flag product docs

16 **Force non-streaming requests via model.streaming** NEW

65

Adds `model.streaming: false` config key to force non-streaming requests for the whole session (parent and subagents), as an escape hatch for self-hosted OpenAI-compatible servers with broken tool-call paths such as vLLM with `--tool-call-parser qwen3_xml`.

— Named config key and example server flag, no worked example product docs

17 **Turn budget warning checkpoint** NEW 65

Adds `agent.budget_warning_ratio` config key (strictly between 0 and 1, off by default) to append a one-time model-visible checkpoint notice after a configurable fraction of the turn budget is consumed, with Dispatcher-owned Kanban workers defaulting to a 90% threshold.

— Named config key with default behavior, no example product docs

18 **Skill environment variable forwarding over SSH** NEW 65

Supports forwarding skill-declared environment variables over SSH via OpenSSH `SendEnv`, with variable names listed under `terminal.env_passthrough`; requires `AcceptEnv` in `/etc/ssh/sshd_config` on the server.

— Names config key and server-side requirement, no example product docs

19 **Background token keepalive interval control** NEW 65

Adds `keepalive_interval_seconds` config key to cap the background token-refresh tick interval for long-running gateway and dashboard processes; set to `0` to disable the keepalive entirely.

Disable the background token keepalive for a gateway process that manages its own credential lifecycle.

```yaml
keepalive_interval_seconds: 0  # disables the keepalive
```

— Named config key with example disabling it product docs

20 **Uninstall orphaned skill entries** NEW 60

Adds `hermes skills uninstall <name>` shell command to remove missing-directory (orphaned) skill entries detected during install checks.

Remove a stale skill entry whose backing directory no longer exists, as flagged by the orphaned install check.

```
$ hermes skills uninstall <name>
```

— Exact runnable command given product docs

21 **Masked sudo password prompts in parent sessions** NEW

60

Supports masked password prompts for interactive `sudo` commands in parent sessions, including literal absolute or quoted executable paths and `env` prefixes with ordinary options and assignments (e.g. `env -u UNUSED /usr/bin/sudo id`); passwordless sudo requires no prompt.

— Names supported invocation patterns but no runnable example product docs

22 **New provider and model routing targets** NEW

60

Adds new provider/model targets including `alibaba-cn`, `alibaba-coding-plan-cn`, `alibaba-token-plan`, `nebius`, `nebius-token-factory`, `nebius-tf`, `tokenfactory`, `tencent-tokenplan`, `tokenplan`, `tencent-lkeap`, `router`, `ramp-router`, and `ramp`.

— Names every new target but no usage example product docs

THINNER COVERAGE BELOW

23 **Context token usage breakdown reporting** NEW

55

Adds context token usage reporting with a `~` prefix on estimates, broken down by category, free-space, skill, and toolset; `/context` reports the selected source.

— Names /context command but no worked example product docs

24 **Skill hashing ignores generated runtime caches** IMPROVED

55

Generated runtime caches (`__pycache__/`, `.pytest_cache/`, `.mypy_cache/`, `.ruff_cache/`, and `.pyc` files) are excluded from skill hashes, so running a skill's helper script no longer marks it user-modified or hides it from `hermes skills list-modified`.

— Names exact excluded paths but no example command product docs

25 **Background review frequency and reasoning controls** IMPROVED

55

Same-model background reviews always inherit the parent conversation's `reasoning_effort` setting, so `auxiliary.background_review.reasoning_effort` has no effect when the review model matches the parent. Adds `memory.nudge_interval` and `skills.creation_nudge_interval` as separate controls to reduce review frequency without changing the main conversation's reasoning effort.

— Named config keys but no example or default values product docs

26 **Git commands default to SSH batch mode** BREAKING 55

Hermes internal Git commands now default to `ssh -o BatchMode=yes`, causing unknown host keys, passwords, and passphrase-protected keys to fail rather than open a terminal prompt; `GIT_SSH_COMMAND` still takes full precedence for custom identity or transport commands.

– Describes exact flag and override but no migration example product docs

27 **hermes chat oneshot and query-file options** NEW 55

Adds `--oneshot` flag to `hermes chat` to answer a query and exit instead of seeding an interactive session, and `--query-file` option to read a query from a file.

– Named flags but no worked example product docs

28 **Auth priority and refresh subcommands** NEW 55

Adds `hermes auth priority` subcommand to reorder credential fill-first order and `hermes auth refresh` subcommand to refresh one OAuth credential and clear its cooldown.

– Two exact subcommands, no worked example shown product docs

29 **Backslash escaping for dots in config keys** NEW 55

Adds backslash escaping for literal dots in `hermes config set/get/unset` key paths, enabling addressable model IDs like `grok-4.6` and Matrix room IDs containing dots.

– Named commands and example IDs but no full worked example product docs

30 **Warning on importing an older backup** IMPROVED 50

Adds an explicit warning when `hermes` imports an older backup over newer work, reporting session and message count deltas (e.g. `state.db: 12 session(s) / 8912 message(s) -> 3 / 24`).

– Concrete example output but no command shown product docs

31 **Automatic retry after messaging flood-control penalties** IMPROVED 45

Adds automatic retry after flood-control rate-limit penalties (e.g. Telegram) without requiring a reconnect or restart; a restart during the penalty adopts the stored reply without spending a retry attempt, retries retain the original bot profile, chat, and thread, and a rate-limit recovery prefix warns that earlier chunks may already have arrived.

– Mechanism described but no config key or example product docs

32 **Skill batch summaries report applied results** IMPROVED 40

Batch summaries for skill operations now report applied results — including supporting-file writes/removals and skill deletions — rather than assuming requested writes succeeded; staged and rolled-back batches are excluded from these summaries.

— Describes behavior fix with no example or command `product docs`

---

33 **Numeric keypad newline in Kitty protocol terminals** NEW

30

Supports inserting a newline via the numeric keypad `Enter` in terminals using the Kitty keyboard protocol, including next to a collapsed paste.

— Narrow terminal compatibility fix with no example `product docs`

---

34 **No conversation reset on inactivity or daily boundary** IMPROVED

30

Gateway conversations no longer reset after inactivity or at a daily boundary; legacy reset-policy overrides and reset-timer environment variables are now ignored.

— States behavior change with no config surface or example `product docs`

---

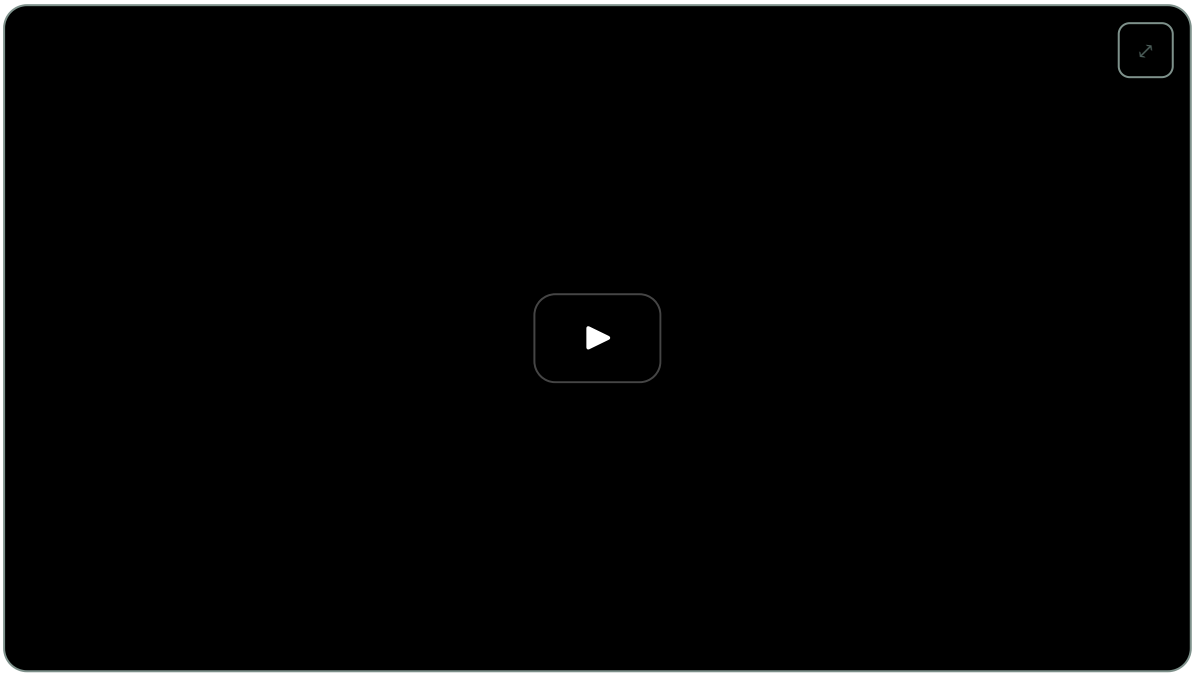
35 **Full command description hover on autocomplete** NEW

25

Adds full command description hover on slash autocomplete rows in the desktop app.

— Brief UI note with no further detail `v2026.9.7`

└──▷ ALSO FROM THESE RELEASES



└──▷ BREAKING ON UPGRADE

! `platforms.<name>.enabled: false` in `config.yaml` now prevents the adapter from starting even when matching credentials (e.g. `WEIXIN_TOKEN`, `TELEGRAM_BOT_TOKEN`) are

present in `.env`; previously, credential presence alone re-enabled twelve platforms regardless of that key.

! `platforms.<name>.enabled: false` is now authoritative: credentials in `.env` for the 12 platforms (Weixin, WhatsApp Cloud, Home Assistant, Email, SMS, DingTalk, Feishu, WeCom, WeCom callback, BlueBubbles, QQ Bot, Yuanbao) no longer re-enable the adapter when this key is set. If you relied on credentials overriding an explicit disable, the gateway now logs one WARNING per affected platform at startup instead of silently starting the adapter.

WAS THIS USEFUL? 👍 👎

Agno (formerly Phidata)

Feedback ⓘ 1 RELEASE · 2026-09-08

NOTES ↗ RANK

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

> Agno v3.0.7 adds a public-serving surface for Agents, Teams, MCP and Workflows, introduces page-based knowledge storage with a bounded read-only filesystem toolkit for agents, and adds an AIMLAPI model integration, alongside breaking changes to the Knowledge constructor and page-search defaults.

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

01 **Bounded read-only page filesystem toolkit** NEW 93

Adds PageFileSystem(knowledge=...) toolkit exposing files.tools() for bounded, read-only, sync/async page access — lazy reads against pinned revisions, scoped metadata listing, and literal grep — without shell execution or write access.

Give an agent bounded, read-only access to page content without exposing shell execution.

python

```python
from agno.knowledge.page import PageFileSystem, Knowledge
from agno.agent import Agent

fs = PageFileSystem(knowledge=Knowledge(name="docs"))

agent = Agent(
    tools=fs.tools(),
    description="Reads documentation pages on demand.",
)
agent.print_response("Summarise the getting-started page.",
stream=True)
```

— Named toolkit and method with a runnable code example. v3.0.7

02 **AIMLAPI model integration** NEW 88

Adds AIMLAPI model integration with a fixed analytics-only attribution header set inspectable as `agno.models.aimlapi.AIMLAPI_HEADERS` and overridable per model via `default_headers`; defaults to model `gpt-5.6-terra`.

Use AIMLAPI as the model provider, overriding the default attribution headers on a per-model basis.

```python
from agno.models.aimlapi import AIMLAPI
from agno.agent import Agent

agent = Agent(
    model=AIMLAPI(
        default_headers={"X-Custom-Key": "my-value"},
    ),
    description="Agent backed by AIMLAPI (defaults to gpt-5.6-terra).",
)
agent.print_response("Hello!", stream=True)
```

— Named module, header constant and override flag with runnable example. v3.0.7

03 **Breaking changes to Knowledge constructor and page search** 88

BREAKING

`Knowledge` constructor arguments are now keyword-only, so positional calls such as `Knowledge('docs')` must become `Knowledge(name='docs')`. `content_db` is now the preferred spelling of the former `contents_db` field on `Knowledge`; `contents_db` remains a read/write alias but passing both keywords requires the same object. Page search no longer silently forces `ef_search=200`, instead honouring the configured `PgVector` HNSW `ef_search` — deployments relying on the implicit 200 must set it explicitly, and page indexes built with the implicit `ef_construction=64` from interim main builds require an operator-managed rebuild.

— Exact field names and required migration steps given, no code sample. v3.0.7

04 **PublicSurface for exposing Agents, Teams, MCP and Workflows** 72

NEW

Adds `PublicSurface` for publicly serving selected Agents, stateless MCP, and authenticated durable-sync Workflows, with shared quotas, CORS, request/output bounds, and internal-service authentication. Extended in the same release to serve explicitly selected Teams with REST and gzip support, sharing the same Agent admission, execution, and output limits.

— Concept and scope described but no runnable example or config shown. v3.0.7

05 **Page-based knowledge storage system** NEW 67

Adds `agno.knowledge.page` with `Knowledge(page_search=PageSearchConfig(...))` for typed, transaction-local planner controls, installable via the new `agno[pages]` extra.

— Names config class and install extra but no usage example. v3.0.7

06 **Expanded arguments for agent dependency callables** IMPROVED

64

Agent dependency callables can now receive `run_input` and `session` alongside the existing `agent` and `run_context` arguments; resolution runs before pre-hooks and reuses successful values on model retries.

— Names arguments and timing but no example call. v3.0.7

THINNER COVERAGE BELOW

07 **StepError reporting in Workflow streaming** IMPROVED 54

Workflows now report failed steps as `StepError` (carrying step identity) in both console streaming printers, distinct from successful `StepOutput` results.

— Names the new type but no example of handling it. v3.0.7

└──▷ BREAKING ON UPGRADE

! Knowledge constructor arguments are now keyword-only: positional calls such as Knowledge('docs') must become Knowledge(name='docs').

! `content_db` is now the preferred spelling of the former `contents_db` field on Knowledge; `contents_db` remains a read/write alias but passing both keywords requires the same object.

! Page search no longer silently forces `ef_search=200`; it now honours the configured `PgVector` HNSW `ef_search`, so deployments that relied on the implicit 200 must set it explicitly. Page indexes built from interim main builds that carry an implicit `ef_construction=64` require an operator-managed rebuild.

WAS THIS USEFUL? 👍 👎

OpenAI Agents SDK

Feedback ⓘ 1 RELEASE · 2026-09-08 NOTES ↗

RANK

OpenAI Agents SDK is an open-source framework that provides tools, handoffs, guardrails, and tracing for agentic applications.

> OpenAI Agents SDK's v0.22.1 window adds tool-level input/output guardrails for function tools and MCP servers with customizable blocked messages, host-environment isolation and Docker labeling for sandbox sessions, and image-result support in web search, alongside smaller additions to tool typing and voice transcription.

└──▷ WHAT SHIPPED · 6 FEATURES most completely described first ? what's the number?

01 Sandbox host-environment isolation and Docker labels NEW

96

Adds `inherit_host_environment=False` on `agents.sandbox.sandboxes.unix_local` to pass only a conservative allowlist of host variables (instead of the full host environment) to sandbox commands, with `host_environment_allowlist` accepting a custom collection to replace the default set (`PATH` , `LANG` , `LC_ALL` , `TZ` , `SSL_CERT_FILE` , `CI` , and others). Adds `labels` support for Docker sandbox sessions, passing key-value pairs to Docker at container creation, storing them in `DockerSandboxSessionState` , and verifying label consistency on resume(...) — raising `ValueError` if any persisted label no longer matches.

Restrict a Unix-local sandbox session to a minimal set of host environment variables to reduce accidental secret leakage into agent subprocesses.

```python
session = await agents.sandbox.sandboxes.unix_local.create(
    inherit_host_environment=False,
    host_environment_allowlist={"PATH", "LANG",
"SSL_CERT_FILE"},
)
```

Tag Docker sandbox containers with application metadata so your orchestration layer can identify and manage them by owner and environment.

```python
session = await client.create(
    labels={
        "com.example.owner": "agents-sdk",
        "com.example.environment": "development",
    },
)
```

— Named params, classes and two runnable examples v0.22.1

02 **Tool guardrails for function tools and MCP servers** NEW

93

Adds `tool_input_guardrails` and `tool_output_guardrails` parameters to local MCP server classes, letting you attach server-wide guardrails that apply to every MCP tool after filtering — input guardrails can block the call and supply replacement content, output guardrails inspect the converted result before it is returned to the model. Guardrails attached directly to local MCP servers apply to every tool exposed by that server, and input guardrails run before and output guardrails after every guarded function-tool invocation, including local MCP tools. Adds `RunConfig.output_guardrail_blocked_message` to set a custom placeholder string or synchronous formatter, receiving `OutputGuardrailBlockedMessageArgs` (SDK default, guardrail name, agent, and active run context), when an output guardrail blocks a tool response.

Prevent MCP tools from being called with arguments containing secrets by attaching a server-wide input guardrail to a local MCP server.

```python
from agents.decorators import tool_input_guardrail
import json
from agents import ToolGuardrailFunctionOutput

@tool_input_guardrail
async def block_secret_arguments(arguments):
    tool_arguments = json.loads(arguments)
    if "secret" in tool_arguments:
        return ToolGuardrailFunctionOutput(
            reject_content="Remove secrets before calling this
MCP tool."
        )
    return ToolGuardrailFunctionOutput(allow=".")

# Pass to your local MCP server
server = MCPServerStdio(
    ...,
    tool_input_guardrails=[block_secret_arguments]
)
```

Supply a custom policy message when an output guardrail blocks a function-tool result, using a synchronous formatter that includes the guardrail name.

```python
from openai_agents import RunConfig,
OutputGuardrailBlockedMessageArgs

def my_blocked_formatter(args:
OutputGuardrailBlockedMessageArgs) -> str:
    return f"Output blocked by policy: {args.guardrail_name}."

run_config = RunConfig(
    output_guardrail_blocked_message=my_blocked_formatter
)
```

— Named params and classes with two runnable code examples v0.22.1

---

03 **Image results and content-type controls in web search** NEW

78

Adds `search_content_types` parameter supporting `"image"` and `"text"` values to control whether web search returns images, text results, or both, plus `image_settings.max_results` to request a specific number of image results and `image_settings.caption` to request short descriptions when available. When

`"image"` is included, the SDK automatically requests `web_search_call.results`, stored on the `web_search_call` item in `RunResult.raw_responses`, with fields `image_url`, `source_website_url`, `thumbnail_url`, and `caption`.

— Named params and result fields but no code sample `v0.22.1`

THINNER COVERAGE BELOW

04 **Annotated variadic *args/**kwargs tool parameters** NEW

58

Supports `Annotated[..., Field(...)]` constraints on variadic `*args` and `**kwargs` tool parameters, with scalar positional values annotated as `*args: T` and homogeneous tuple values as `*args: tuple[T, ...]`.

— Precise typing syntax named, no runnable example given `product docs`

05 **MCP list_tools caching returns detached copies** IMPROVED

44

Documents that when caching is enabled, `list_tools` results contain detached copies of cached tool definitions (including nested input schemas), so mutating a returned tool or a tool received by a dynamic filter callback does not affect the server's cached schema or future `list_tools` results.

— Behavior explained but no example or exposed flag `product docs`

06 **Streamed transcription options for voice pipelines** NEW

20

Exposes streamed transcription options for voice pipelines.

— One-line description with no named option or example `v0.22.1`

WAS THIS USEFUL? 👍 👎

holmesgpt

Feedback ⓘ 1 RELEASE · 2026-09-08 NOTES ↗ RANK

SRE Agent - CNCF Sandbox Project

> HolmesGPT 0.41.0 lets skill repositories refresh live without restarting the agent and hardens its relay authentication with cached API key fetching and automatic JWT renewal.

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

01 Live-refresh of git-synced skills via skill_repos NEW

69

Introduces a first-class `skill_repos` configuration that lets Holmes pick up updates to git-synced skill repositories live, eliminating the need to restart Holmes when skill repositories change.

— Names config key and behaviour but no restart/refresh mechanics. 0.41.0

THINNER COVERAGE BELOW

02 Relay authentication resilience improvements IMPROVED

40

Holmes now fetches the Supabase API key from the relay with caching and a local fallback, improving resilience when the relay is temporarily unreachable, and automatically refreshes the realtime JWT before it expires to maintain uninterrupted realtime connections.

— Describes behaviour but no config keys or endpoints given. 0.41.0

WAS THIS USEFUL? 👍 👎

LangChain

Feedback ⓘ 1 RELEASE · 2026-09-08 NOTES ↗ RANK

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

> langchain-openai 1.6.1 adds asynchronous tool execution and a new `configuration_update` option for OpenAI models.

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

01 **Async tool execution for OpenAI integration** NEW 45

The OpenAI integration now supports async tools, allowing tool calls to be executed asynchronously.

— States capability but no usage example or mechanism detail. langchain-openai==1.6.1

02 **`configuration_update` for OpenAI models** NEW 30

langchain-openai 1.6.1 adds `configuration_update` for OpenAI models, per the release summary; no further detail on its behavior or parameters is provided.

— Named config key only, no explanation of behavior or usage. langchain-openai==1.6.1

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Agentic Infrastructure

HeyGen HyperFrames

Feedback ⓘ 1 RELEASE · 2026-09-07 NOTES ↗

RANK

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

> HyperFrames v0.8.31 adds a linting rule to catch mismatched media references before rendering.

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

01 **Lint rule for mismatched media src types** NEW 55

A new `lint` rule flags `video` / `img` elements whose `src` attribute points at an audio file, catching mismatched media type references before render.

– Names the rule and elements checked but no usage example. v0.8.31

WAS THIS USEFUL? 👍 👎

BerriAI LiteLLM

Feedback ⓘ 1 RELEASE · 2026-09-06 NOTES ↗

RANK

LiteLLM is an AI gateway that calls over 100 model APIs in OpenAI format with cost tracking, guardrails, load balancing, and logging.

> LiteLLM v1.100.0 expands provider and API coverage with Bing Search grounding and native Vertex AI Interactions support, adds enterprise routing and security controls for RAG ingestion and Azure AI Foundry auth, and removes the prompt_token_calculator utility in a breaking change.

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

01 **Security controls on RAG ingest endpoint** IMPROVED 60

Enforces vector-store upload security controls on the `/v1/rag/ingest` endpoint.

– Names exact endpoint, mechanism only briefly described v1.100.0

THINNER COVERAGE BELOW

02 **prompt_token_calculator utility removed** BREAKING 55

The `prompt_token_calculator` utility is deleted; any code importing or calling it will break.

– Names exact removed utility and impact, no migration path given v1.100.0

03 **Bing Search grounding provider** NEW 50

Adds `bing_grounding` as a search provider via Grounding with Bing Search on the proxy.

– Names the config key but no usage mechanism detail v1.100.0

04 **Entra ID / OAuth on Azure AI Foundry routes** NEW 50

Adds Entra ID / OAuth authentication support on every Azure AI Foundry route.

– Names auth method and scope, no setup steps v1.100.0

05 **Per-team New Relic trace routing** NEW 45

Adds per-team New Relic trace routing via team callbacks.

– Names the callback mechanism but no config example v1.100.0

06 **Error-code drilldown on caching page** IMPROVED 45

Adds error-code drilldown for failed requests on the caching page in the UI.

— Names UI location, limited mechanism detail v1.100.0

07 **Complexity router classifier context bounding** IMPROVED 45

Bounds the complexity router classifier context block (not each turn individually), giving finer control over routing decisions.

— Explains before/after behavior change, no config surface v1.100.0

08 **Streamed response record-and-replay for testing** NEW 45

Adds chunk-for-chunk record-and-replay of streamed provider responses for E2E testing.

— Describes mechanism but no invocation details v1.100.0

09 **Gemini Family auto-router preset** NEW 40

Adds a Gemini Family auto-router preset in the UI.

— UI location named but behavior unexplained v1.100.0

10 **Vertex AI Interactions API support** NEW 35

Adds native Vertex AI Interactions API support.

— Bare name, no mechanism or usage given v1.100.0

└──▷ **BREAKING ON UPGRADE**

! The `prompt_token_calculator` utility is deleted; any code importing or calling it will break.

WAS THIS USEFUL?  

iOfficeAI OfficeCLI

Feedback ⓘ 1 RELEASE · 2026-08-31

NOTES ↗ RANK

OfficeCLI is the first and best Office suite purpose-built for AI agents to read, edit, and automate Word, Excel, and PowerPoint files. Free, open-source, single binary, no Office installation required.

> OfficeCLI v1.0.146 adds detection for numeric display overflow in Excel spreadsheets.

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

01 **Numeric overflow reporting for .xlsx files** NEW 43

OfficeCLI now reports numeric display overflow in `.xlsx` spreadsheets, flagging cells where numbers are too wide to render correctly.

— Describes behavior but no command or config to invoke it v1.0.146

WAS THIS USEFUL? 👍 👎

OpenRouter

Feedback ⓘ 1 RELEASE · 2026-08-17 NOTES ↗ RANK

OpenRouter is a routing gateway that exposes hundreds of models from many providers behind one OpenAI-compatible API with failover and unified billing.

> OpenRouter's latest snapshot introduces a full analytics suite: a queryable Analytics API alongside a dashboard with team-level spend breakdowns, saveable charts, and log drill-down.

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

01 **Analytics API and dashboard drill-down** NEW 50

OpenRouter adds an Analytics API so spend and usage data can be queried from the terminal or scripts. The analytics dashboard gains per-model spend breakdowns scoped to your team, saveable charts that persist across sessions, and click-through from any chart bar directly into the underlying request logs.

— Names an Analytics API but no endpoint, params, or command shown. snapshot-20260908

WAS THIS USEFUL? 👍 👎

ModelScope MS-SWIFT

Feedback ⓘ 1 RELEASE · 2026-09-08

NOTES ↗ RANK

MS-SWIFT provides training, fine-tuning, reinforcement learning, evaluation, and deployment workflows for language and multimodal models.

> MS-SWIFT v4.5.3 adds a Kunlun XPU inference backend, multimodal OPD distillation, per-group Muon learning rates, and expands supported model families to include Qwen3.8-Flash-Next, DeepSeek-V4-Pro-0813, and five other new models.

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

01 **New model family support across seven models** NEW 75

Adds support for Qwen3.8-Flash-Next (a 125B multimodal MoE with ~6B activated parameters per token, 262K native context, and Megatron training support), deepseek-ai/DeepSeek-V4-Pro-0813, inclusionAI/Ling-3.0-tiny and inclusionAI/Ling-3.0-flash (including agent templates), XHToken/Spark-X2.5, lmms-lab/LLaVA-OneVision-2-8B-Instruct, iic/UEmbed-2B, and Tencent-Hunyuan/WeMM-Embedding (2B/4B/9B, including VLLM inference backends).

— Enumerates model names and specs but no usage commands v4.5.3

02 **Persistent dataloader workers enabled by default** IMPROVED 65

Sets `dataloader_persistent_workers` to `True` by default, eliminating dataloader cold-start overhead on every eval.

— Names config key, default value, and the effect v4.5.3

THINNER COVERAGE BELOW

03 **vllm_kunlun inference backend for Kunlun XPU** NEW 55

Adds the `vllm_kunlun` inference backend, enabling rollout and deploy workflows to run on Kunlun XPU hardware.

— Names backend and hardware target but no setup steps v4.5.3

04 **Multimodal OPD distillation with separate teacher/student vision inputs**

IMPROVED 55

OPD distillation (GRPO) now supports separate vision inputs for the teacher and student models, enabling multimodal distillation workflows.

— Explains mechanism but gives no config example v4.5.3

05 **Automatic last-checkpoint symlink** IMPROVED 50

A `last-checkpoint` symlink is now maintained automatically when saving checkpoints, making resume easier.

— Names the symlink but no resume command shown v4.5.3

06 **Per-parameter-group learning rates in Megatron-SWIFT Muon optimizer**

IMPROVED 45

The Megatron-SWIFT Muon optimizer now supports different learning rates for different parameter groups.

— States the change but no configuration details v4.5.3

07 **Removed gradio dependency** DEPRECATED 25

Removes the `gradio` dependency from requirements.

— Bare statement with no explanation of impact v4.5.3

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

EverMind AI EverOS

Feedback ⓘ 1 RELEASE · 2026-09-08 NOTES ↗

RANK

EverOS provides a local-first memory layer that stores and evolves context for AI agents.

> EverOS v1.3.1 adds LLM-guided multi-round memory retrieval, cascade quiesce endpoints for safe snapshotting, expanded LLM/decider configuration, a unified benchmark harness, and a breaking default timeout on OME strategy runs.

└──▷ WHAT SHIPPED · 5 FEATURES most completely described first ? what's the number?

01 **OME strategy timeout enforcement** BREAKING 91

OME strategy attempts now enforce a 1,800-second default wall-clock timeout, tunable via `EVEROS_OME_RUN_TIMEOUT_SECONDS`, and follow the existing retry/dead-letter path on timeout. Strategies that legitimately run longer than 1,800 seconds will now time out and enter the dead-letter path on upgrade; set `EVEROS_OME_RUN_TIMEOUT_SECONDS` to a higher value or `0` / `off` before upgrading to preserve prior behavior.

— Exact env var, default value, and migration steps provided. v1.3.1

02 **Cascade snapshot control endpoints** NEW 88

`POST /api/v1/cascade/quiesce` and `POST /api/v2/cascade/quiesce` drain the Markdown-to-index queue and halt the cascade subsystem until restart, with startup switches to disable all cascade work or only the filesystem watcher.

Quiesce the cascade subsystem before taking a read-only snapshot of the index, then restart the service to resume normal ingestion.

```
$ curl -X POST https://<host>/api/v2/cascade/quiesce
```

— Two versioned endpoints with precise behavior and an example call. v1.3.1

03 **LLM-guided multi-round memory retrieval** NEW 85

`POST /api/v2/memory/search` now accepts `method = "llm_multiround"` for iterative, multi-round episode retrieval that fuses BM25 and vector candidates with RRF each round and returns a bounded final context without a cross-encoder.

Run iterative multi-round memory retrieval when a single-pass search misses context spread across many episode blocks.

```
$ curl -X POST https://<host>/api/v2/memory/search \
  -H 'Content-Type: application/json' \
  -d '{"query": "What did Alice say about the project
deadline?", "method": "llm_multiround"}'
```

— Exact endpoint, method value, mechanism, and runnable curl example. v1.3.1

04 **LLM and decider configuration options** NEW 65

Adds a `[decider]` config section to independently configure the model, endpoint, timeout, request extras, retry policy, and loop tuning for the multi-round retrieval decider, with empty connection fields inheriting from `[llm]`. Also adds `[llm]` request timeout and provider-specific SDK argument settings, retaining the previous 60-second default.

— Named config sections and keys but no file path or example. v1.3.1

05 **Unified benchmark harness** NEW 60

A unified benchmark harness covers LoCoMo, LongMemEval, EverMemBench, and SubtleMemory under one runner with explicit dataset configs, resumable stage artifacts, deterministic run identities, shared IR metrics, and decider preflight checks.

— Names benchmarks and features but gives no runnable command. v1.3.1

└──▷ BREAKING ON UPGRADE

! OME strategies that legitimately run longer than 1,800 seconds will now time out and enter the dead-letter path on upgrade; set `EVEROS_OME_RUN_TIMEOUT_SECONDS` to a higher value or `0` / `off` before upgrading to preserve prior behavior.

WAS THIS USEFUL? 👍 👎

okf-memory OKF Agent Memory

Feedback ⓘ

2 RELEASES · 2026-09-05 → 2026-09-08 NOTES ↗ RANK

Git-native persistent memory for AI coding agents. 2 with sub-300µs in-memory BM25 search, embedded MCP server, and progressive disclosure.

> OKF Agent Memory debuted with a Git-native memory engine, sub-300µs BM25 search, and an embedded stdio MCP server that plugs into Claude Code, Cursor, and Codex, then followed with a broad security-hardening pass adding workspace confinement, input sanitization, and a continuous adversarial audit framework.

└──▷ WHAT SHIPPED · 11 FEATURES most completely described first ? what's the number?

01 **Workspace confinement and input sanitization for MCP server**

NEW 93

The `OKF_MCP_ROOT` environment variable sets a canonical root directory for the stdio MCP server; dynamic validation rejects any `bundle` argument escaping that root (e.g. `bundle="/etc"` or `bundle="../../outside"`), and out-of-root `bundle` tool calls are denied with a JSON-RPC tool error. Concept persistence endpoints (`SaveConcept` , `UpdateParentIndex` , `AppendLogEntry`) enforce canonical bundle containment, rejecting concept IDs with `..` segments or absolute/escaping paths. `LoadBundle` inspects symlinks and rejects any resolving outside the canonical bundle root or pointing to directories or non-markdown files. `sanitizeConceptMetadata` rejects newline characters and `---` frontmatter delimiters in the `type` , `title` , `description` , and `actor` fields, blocking YAML injection and forged verification states. The four defense-in-depth choke-points are documented in `knowledge/architecture/security-boundaries.md` , with operational conventions in `knowledge/convention/mcp-agent-safety.md` .

Pin the MCP server to a specific project root so agent tool calls cannot escape it, even if a malicious `bundle` path is supplied.

```
$ OKF_MCP_ROOT=/path/to/project ./bin/okf mcp knowledge
```

— Names env var, functions, files and exact rejected inputs v0.1.3

02 **Embedded MCP server for agent connection** NEW 88

`okf mcp knowledge` launches a native Model Context Protocol server over `stdio` , connecting AI coding agents (Claude Code, Cursor, Codex) to the local knowledge graph without an external service. Agents are wired in via a config entry (e.g. `claude_desktop_config.json`) pointing the `okf` binary at `mcp` with the knowledge directory as an argument.

Wire an existing project's knowledge directory into Claude Desktop so the agent can query and write persistent memory across sessions.

```json
{
  "mcpServers": {
    "okf-memory": {
      "command": "/usr/local/bin/okf",
      "args": ["mcp", "/path/to/project/knowledge"]
    }
  }
}
```

— Names exact command and config wiring; mechanism (stdio) given `v0.1.0`

---

03 **Knowledge validation with drift and staleness gating** IMPROVED

86

`okf validate knowledge --strict --drift` checks OKF v0.2 bundle conformance, bidirectional graph connectivity, and description drift across the `knowledge/` corpus. v0.1.3 adds a `--stale` flag so CI pipelines can fail the build if any concept has passed its `stale_after` date, gating on refreshed memory before passing.

Fail CI if any concept has passed its `stale_after` date, ensuring agents never act on rotted memory.

```
$ ./bin/okf validate knowledge --strict --stale
```

— Exact command and flags with CI usage example `v0.1.3` `v0.1.0`

---

04 **BM25 concept search** NEW

84

`okf search` performs sub-300µs in-memory BM25 concept retrieval against the knowledge graph with zero vector-database dependencies and zero API cost, e.g. `okf search "OAuth2 authorization" knowledge` to check for existing concepts before writing new ones.

Search the knowledge graph for relevant concepts before writing new ones, following the Search-Before-Write principle to avoid duplication.

```
$ okf search "OAuth2 authorization" knowledge
```

— Runnable command with concrete latency figure `v0.1.0`

---

05 **Continuous adversarial security audit framework** NEW

78

`make audit-security` runs `gosec` and `govulncheck` for automated security scanning of the codebase. `docs/SECURITY_AUDIT.md` defines a standardized 4-area adversarial checklist (Filesystem/CWE-22, MCP/Agent Interfaces, DoS/Bounds, Data Integrity) for Security Reviewer Agents, `docs/RELEASE_PLAYBOOK.md` adds a mandatory Step 2 Security & Adversarial Audit Gate for every release, and a Google Jules prompt template enables continuous, autonomous daily repository security audits.

Run the full automated security audit against the codebase to catch filesystem and dependency vulnerabilities before a release.

```
$ make audit-security
```

— Names exact make target, tools, doc files and audit gate step `v0.1.3`

---

06 **Bootstrap scaffolding command** NEW

78

`okf bootstrap /path/to/project --name` scaffolds the full OKF Agent Memory stack — `knowledge/` , `.agents/skills/okf-memory/` , `AGENTS.md` , and `Makefile` — into a target project in one command.

Scaffold the full agent memory stack into a new service repo so AI coding agents immediately have structured, auditable memory.

```
$ okf bootstrap /path/to/my-service --name "My Service"
```

— Exact command with concrete scaffolded paths `v0.1.0`

---

07 **Concept authoring, inspection and init commands** NEW

77

`okf create` authors new OKF v0.2 concept files with `--type` , `--title` , and `--desc` flags, automatically updating `log.md` and `index.md` bookkeeping. `okf show <concept> knowledge --json` inspects a concept and its graph relationships as JSON, `okf update <concept> knowledge --desc` revises an existing concept's description in-place, and `okf init` initializes a bare OKF v0.2 bundle (`index.md` , `log.md`) in any target directory.

— Names every flag and file but no runnable example given `v0.1.0`

---

08 **Trust and provenance tiers in concept frontmatter** IMPROVED

65

Frontmatter fields `verified: human:lead@...` vs. `generated: agent/...` formally distinguish authoritative human decisions from agent drafts. v0.1.3 adds trust

chronology validation so `verified.at` cannot precede `generated.at`, preventing AI agents from self-authorizing verified status.

— Names exact fields and the chronology rule enforced v0.1.3 v0.1.0

THINNER COVERAGE BELOW

09 **Progressive disclosure context reduction** NEW 58

Hierarchical `index.md` link graphs let agents fetch 300-token atomic concepts instead of 20k-token monolith files, reducing agent context-window consumption by up to 94%.

— Gives concrete token numbers but no command to invoke it v0.1.0

10 **Lightweight Go binary runtime** NEW 47

Ships as a single compiled Go binary with a process cold-start under 4 ms and an RSS footprint under 15 MB, enabling high-frequency agent tool-calling loops.

— Concrete perf numbers but nothing actionable to run v0.1.0

11 **Git-native Markdown storage with audit trail** NEW 43

All agent memory is stored as plain Markdown files with YAML frontmatter under `knowledge/`, making every change auditable via `git diff` and standard PR review.

— Describes mechanism but no command or config surface v0.1.0

WAS THIS USEFUL? 👍 👎

Volcengine OpenViking

Feedback

2 RELEASES · 2026-09-08

NOTES ↗ RANK

OpenViking is a context database that manages agent memory, knowledge retrieval, and skills.

> OpenViking shipped resource-level ACL with group-based permissions and an account-level switch, transactional cp/mv that migrates vector records without re-parsing, and pagination plus tag filtering across ls, tree, write, grep and glob, alongside a new restricted Python DSL session-extraction default, Feishu wiki ingestion unification, and a Web Studio search panel and Agent Experience module.

└──▷ WHAT SHIPPED · 17 FEATURES most completely described first ? what's the number?

01 Resource-level ACL with groups and inheritance

BREAKING

90

OpenViking adds resource-level ACL for `viking://resources/...` with principals `user:{user_id}`, `group:{group_id}`, and `user:*` and levels `read` / `write` / `manage`, with vector retrieval filtering hits by effective permission. It adds an account-level ACL switch via `{"acl":{"enabled":true|false}}` in account settings, `--acl-enabled true|false` in the CLI, and the `PATCH /api/v1/admin/accounts/{account}/settings` endpoint; disabling fully restores pre-ACL sharing behavior. It also adds `restricted inheritance mode` to the ACL system for tighter permission propagation control. The derived-state field `acl_enabled` (boolean) is renamed to `acl_mode` (string): `{"acl_enabled": false}` becomes `{"acl_mode": "none"}` and `{"acl_enabled": true}` becomes `{"acl_mode": "inherit"}`; code checking `acl_enabled` must switch to checking `acl_mode != "none"`.

Enable resource-level ACL for an account so new shared resources record creator `manage` rights and vector retrieval filters by effective permission.

```
$ ov admin set-account-settings acme --acl-enabled true
```

Enable the account-level ACL switch via the HTTP API, for use in automated provisioning pipelines.

```
$ curl -X PATCH
https://<host>/api/v1/admin/accounts/acme/settings -H 'Content-
Type: application/json' -d '{"acl": {"enabled": true}}'
```

— Names endpoints, flags, and exact migration mapping. v0.4.18 v0.4.19

02 Transactional cp/mv with vector record migration

BREAKING

86

OpenViking adds a `POST /api/v1/fs/cp` endpoint and `ov cp` command (with `-r` for recursive copy) that migrate existing vector records without re-parsing. `cp` and `mv` now overwrite existing targets and merge directories instead of failing when the target exists, so `target already exists` can no longer be relied on as a safeguard and overwritten data cannot be recovered.

Recursively copy a resource directory while migrating existing vector records, avoiding re-parsing and re-embedding.

```
$ ov cp -r viking://resources/a viking://resources/b
```

— Exact endpoint, command, flag and behavior change given. v0.4.18

03 **Tag-based filtering across fs commands** BREAKING 84

OpenViking adds `--tags k=v,k2=v2` support to `write`, `ls`, `tree`, `grep`, and `glob` for tag-based AND filtering and optional tag return, without issuing an extra VectorDB query when tags are neither requested nor filtered. This replaces the repeatable `--tag k=v` flag on `ov add-resource` and `ov reindex`, so scripts calling these commands with `--tag` must be updated to `--tags`.

Page through a large resource directory and filter results to a specific team tag, avoiding a full directory scan.

```
$ ov ls viking://resources/docs --offset 100 --limit 50 --tags
team=search,env=prod --fields tags
```

— Exact flag syntax, affected commands, and migration path given. v0.4.18

04 **Pagination and sorting for ls/tree** NEW 75

OpenViking adds `offset` and `limit` pagination parameters to `ls` and `tree` across the HTTP API, CLI, Python/Go/TypeScript SDKs, and MCP, and adds sorting to `ls` in the CLI and MCP.

Page through a large resource directory and filter results to a specific team tag, avoiding a full directory scan.


```
$ ov ls viking://resources/docs --offset 100 --limit 50 --tags
team=search,env=prod --fields tags
```


— Names exact parameters and every surface they apply to. v0.4.18

05 **Safe local schema updates and manual remote migration** BREAKING 75

For local/cuVS backends, OpenViking now performs safe local schema updates on startup: it only appends missing fields and scalar indexes, and never calls drop_index() or rebuilds the vector index. The automatic destructive schema migration on startup has been removed; existing remote VikingDB collections now require operators to manually add ACL fields and corresponding scalar indexes in the console before upgrading.

– Names exact method (`drop_index()`) and manual migration requirement. v0.4.18

06 **Web Studio search panel with five modes** NEW 73

OpenViking adds a Web Studio search panel with five modes — `files`, `find`, `search`, `grep`, and `glob` — with Tab cycling, L0/L1 directory preview, a collapsible interaction panel with persistent state, and structured JSONL rendering.

– Names all five modes and UI mechanics but no exact navigation path. v0.4.18

07 **Deterministic vector record IDs and diagnostics** NEW 68

OpenViking adds deterministic vector record ID generation; `stat` and `read` can resolve files by ID and emit actionable missing-index diagnostics. `ls`, `tree`, and `glob` support optional fields and script-friendly output.

– Names commands and behavior but no exact flags shown. v0.4.18

THINNER COVERAGE BELOW

08 **Multimodal resource reads via VikingBot** NEW 57

OpenViking adds `openviking_multi_read` to VikingBot, returning multimodal content suited for model consumption by OpenViking resource type.

– Names the exact function but no usage example. v0.4.18

09 **Metadata-only resolution in find()** IMPROVED 55

find() now accepts an empty `query` when a `filter` is provided, resolving entirely from metadata storage with `score` fixed at `0`.

– Names the function and field but no example given. v0.4.18

10 **Suspendable Watch creation** NEW 55

OpenViking adds support for `is_active=false` Watch creation, which still runs one initial import but suspends subsequent scheduled cycles.

– Names the exact parameter and its effect. v0.4.18

11 **Feishu wiki and folder ingestion unification** NEW 54

OpenViking adds recursive wiki materialization for Feishu, enabling full wiki-tree ingestion, and unifies Feishu folder and file imports under the Understanding pipeline.

– Names the pipeline but no config or command given. v0.4.19

12 **Restricted Python DSL session extraction protocol** BREAKING 50

OpenViking adds a restricted Python DSL extraction protocol for session data and makes it the default extraction path; existing setups relying on the previous default extraction behaviour will use the new protocol after upgrading.

— No flag or config name given for the DSL protocol. v0.4.19

13 **Ingestion rejects empty-content sources** BREAKING 48

`add_resource` now rejects sources with no content at the ingestion stage; previously they were accepted.

— Names the function and behavior change, no migration steps. v0.4.18

14 **Compile subsystem task tracking and sessions compile skill** NEW 46

OpenViking adds OV-managed external task lifecycle tracking in the compile subsystem, adds a `sessions compile` skill, and adds agent logo localization.

— Names the skill but lifecycle tracking mechanism unspecified. v0.4.19 v0.4.18

15 **Explicit OpenAI-compatible multimodal embedding** NEW 30

OpenViking supports explicit enablement of OpenAI-compatible multimodal embedding.

— Thin one-line mention with no configuration detail. v0.4.18

16 **Agent Experience module in web-studio** NEW 28

OpenViking adds an Agent Experience module to the web-studio UI.

— Bare mention with no functional detail. v0.4.19

17 **Local files in the external queue** NEW 23

OpenViking supports local files entering the external-queue.

— Bare mention with no mechanism or config given. v0.4.18

└──▷ BREAKING ON UPGRADE

! The session extraction protocol now defaults to the restricted Python DSL mode; existing setups relying on the previous default extraction behaviour will use the new protocol after upgrade.

! The ACL derived-state field `acl_enabled` (boolean) is renamed to `acl_mode` (string) in ACL reports and context records: `{"acl_enabled": false}` becomes `{"acl_mode": "none"}` and `{"acl_enabled": true}` becomes `{"acl_mode": "inherit"}`. Update any code that checks `acl_enabled` to check `acl_mode != "none"` instead.

! The `--tag k=v` flag (repeatable) on `ov add-resource` and `ov reindex` is replaced by `--tags k=v,k2=v2` (comma-separated). Scripts calling these commands with `--tag` must be updated.

! `cp` and `mv` now overwrite existing targets and merge directories instead of failing when the target exists. Do not rely on `target already exists` as a safeguard; overwritten data cannot be recovered.

! `add_resource` now rejects sources with no content at the ingestion stage; previously they were accepted.

! Existing remote VikingDB collections require operators to manually add ACL fields and corresponding scalar indexes in the console before upgrading; the automatic destructive schema migration on startup has been removed.

WAS THIS USEFUL? 👍 👎

timgordontg engrim

Feedback ⓘ

2 RELEASES · 2026-08-11 → 2026-09-07 NOTES ↗ RANK

The Universal Cross-Model Episodic Memory Standard. Local-first, project-scoped SQLite memory engine for Google Antigravity, Claude Code, Cursor, Windsurf, and Codex.

> engrim v1.3.0 brings one-command multi-agent setup, lifecycle hooks for Google Antigravity, and a hardened MCP server, while v1.2.0 adds compact action-line transcript logging searchable via a new recall flag.

└──▷ WHAT SHIPPED · 6 FEATURES most completely described first ? what's the number?

01 **Antigravity lifecycle hooks** NEW 80

`engrim hook --agent agy --event boot|stop` adds lifecycle hooks via the new `engrim.adapters.agy` adapter, wiring Google Antigravity's `PreInvocation` and `Stop` events into `~/.gemini/config/hooks.json`.

— Names adapter, events and exact config file path v1.3.0

02 **Unified agent setup command** NEW 75

`engrim setup [--agy|--claude|--cursor|--codex|--all|--dry-run]` configures all supported agent environments in one command, auto-detecting which environments (Antigravity, Claude Code, Cursor, Codex) are installed on the machine.

— Exact flags and behaviour given, no example run shown v1.3.0

03 **Action-line transcript logging** NEW 75

Each state-changing tool call is now recorded as a compact action line (e.g. `[changed] src/engrim/cli.py`, `[ran] ...`) in the transcript log, condensing tool-call payloads from 93 KB to ~10 KB of readable spine. The new `--log` flag on `engrim recall` searches these raw transcript turns alongside curated memory (e.g. `engrim recall -q "release.yml" --log`), and `engrim log --reindex` re-derives searchable action lines from raw turns already on disk without data loss.

— Concrete before/after size and example command given v1.2.0

04 **MCP server for memory tools** NEW 70

`engrim serve --mcp` (alias `engrim mcp`) launches a zero-stdout-contamination JSON-RPC 2.0 stdio MCP server exposing four tools: `engrim_recall`, `engrim_add`, `engrim_context`, and `engrim_review`.

— Command and exposed tool names given, mechanism thin v1.3.0

05 **Origin-agent provenance tracking** NEW 65

Every memory record now carries an `origin_agent` field recording which agent recorded it (`antigravity` , `claude-code` , `cursor` , `cli` , or `user`), surfaced in `engrim context` and `engrim list` output; the `--origin-agent` flag on `engrim add` lets users tag manual insertions explicitly. This underpins cross-agent memory sharing across Google Antigravity, Claude Code, Cursor MCP, and Windsurf on the same codebase without context drift.

— Field and flag named but no example of resulting output v1.3.0

THINNER COVERAGE BELOW

06 **Shared SQLite store env vars** NEW 50

`ENGRIM_DB` and `ENGRIM_PROJECT` environment variables let multiple containers or hosts point at a shared SQLite store while keeping a stable project tag.

— Named env vars but purpose stated briefly v1.3.0

WAS THIS USEFUL? 👍 👎

ralforion OrionBelt Ontology Builder

Feedback ⓘ

10 RELEASES · 2026-08-09 → 2026-09-06 NOTES ↗ RANK

Browser-based ontology workbench for OWL ontologies and SKOS vocabularies. Streamlit + rdflib, no Java, no Protégé.

> OrionBelt's biggest addition this window is a full read-only SPARQL query console (SELECT/ASK/CONSTRUCT/DESCRIBE, CSV/Turtle export, row and time limits, eight worked examples) that gained a prefix table, state persistence and verbatim error display across four releases, alongside a new shortest-path panel for tracing relationships between ontology entities, a major SKOS validation and autofix expansion (5→20 checks plus a 7-class autofixer), and custom language packs for annotation fields; two releases also raised the Python floor to 3.12 and dropped the pyvis dependency.

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

01 **SPARQL query console with read-only guardrails** NEW

100

Adds a SPARQL query console supporting `SELECT`, `ASK`, `CONSTRUCT`, and `DESCRIBE` against the loaded ontology, with results as a table or Turtle, downloadable as CSV or `.ttl`. Refuses `INSERT`, `DELETE`, `LOAD`, `CLEAR`, `DROP`, `SERVICE`, `FROM`, and `FROM NAMED` with an explicit read-only message. Enforces an adjustable per-query row limit and wall-clock time limit, covering `ORDER BY`, `GROUP BY`, `DISTINCT`, and `UNION` queries. Ships eight one-click worked examples (classes and labels, class hierarchy, properties with domain and range, individuals by type, classes with no label, triples per predicate, SKOS concepts with broader terms, full resource description) in a syntax-highlighting editor with line numbers, folding and bracket matching, falling back to plain monospace without `streamlit-ace`. A table of active prefixes and namespaces (excluding rdflib's ~30 defaults) now appears beneath the editor to diagnose empty results, the Plain editor choice, query, row limit and time limit persist across page switches and reloads via local config or `localStorage`, and a failed query now shows the engine's error message verbatim in a code block with a copy button.

Identify every class in the loaded ontology that has no rdfs:label — useful for quickly auditing annotation coverage before publishing.

📍 In the SPARQL page, click the 'classes with no label' worked example, then click Run. The result table lists every class IRI with no rdfs:label triple. Download as CSV for a gap report.

Explore property domain and range declarations across a large ontology without manually browsing each property's editor page.

📍 In the SPARQL page, click the 'properties with domain and range' worked example, then click Run. Results show each property alongside its rdfs:domain and rdfs:range values in a sortable table.

Cap a potentially slow exploratory query — such as a multi-hop path search — so it returns partial results on schedule rather than hanging the app.

📍 In the SPARQL page, set the wall-clock time limit to your desired threshold, write or paste your query (e.g. a three-way join under ORDER BY), then click Run. If the limit is hit, the console reports how many rows were produced before stopping.

When a SPARQL query returns no rows, check the prefix table beneath the editor to confirm which namespace `:` (or any other prefix) actually resolves to before assuming the ontology is empty.

📍 In the SPARQL console, look below the query editor for the prefix/namespace table. Verify that `:` maps to your ontology's base URI (e.g. `http://example.org/ontology#`). If a term such as `:Event` matches nothing, cross-reference the table — the class may live under a different prefix like `gufo:`.

— Names every clause, limit and worked example; fully runnable from this alone. v1.27.3 v1.27.2 v1.26.1 v1.26.0

---

02 **Shortest-path discovery between ontology entities** NEW

98

Adds a shortest-path panel under the Find/Focus row on the Visualization page: pick two entities (classes, individuals, or SKOS concepts) and the graph reports the hop-by-hop chain of links, highlighted on canvas, running over the full rdflib graph rather than just rendered nodes and respecting active display toggles. A Focus on this path mode assembles and renders a path even when intermediate entities are hidden by a filter or the 500-node render cap; highlighted links retain their original colour, weight and dash style with a ring added rather than overpainting, and are now drawn regardless of the node cap with an indicator when only a partial path is rendered.

Find the relationship chain between two ontology classes when investigating how an access role connects to an organization structure.

📍 In the Visualization page, open the Find / Focus row and select the new shortest-path panel. Choose Class: Role in the first picker and Class: Organization in the second. The graph returns the path hop-by-hop — e.g. Class: Role ←subClassOf— Class: Department ←hasDepartment— Class: Organization — and highlights it on the canvas.

Reveal a path that passes through entities hidden by a filter or beyond the 500-node render cap, without manually adjusting view settings.

📍 After the shortest-path panel returns a result that includes entities not currently on the canvas, click `Focus on this path` in the panel. The graph switches to focus mode, assembles the full path past the render cap, and prunes the view to just the nodes and links on that path.

— Worked examples show exact panel navigation and resulting path syntax. `v1.27.2` `v1.27.0`

---

03 **Custom language packs for annotation fields** NEW 90

Adds a `Language Packs` tab under Annotations where custom packs can be created from scratch or copied from an existing pack, edited row-by-row, and imported or exported as JSON, becoming the source for every Language field across all pages and the graph panel. Ships two built-in packs — ISO 639-3 (alpha-3, 228 codes, default) and ISO 639-1 (alpha-2, 184 codes) — projected from one table including historical/special-purpose codes like `grc`, `ang`, `non`, `sux`, `syc`, `und`, `mul`, `zxx`, and `qaa`. Every bare Language text box (Add Annotation, the annotation editor, the graph panel annotate form, Add Concept on the SKOS page) is replaced with a searchable picker showing entries as `eng · English`, with arbitrary BCP 47 tags still typeable directly. The active language pack persists to `~/.orionbelt_ontology_builder/config.json` on desktop and to `localStorage` on the cloud.

Build a short project-specific language list so annotators can only pick the three languages your ontology actually uses, without scrolling through 228 ISO 639-3 codes.

📍 In the sidebar, go to Annotations › Language Packs. Click `New pack`, give it a name, copy rows from the ISO 639-3 built-in pack, delete all but `eng`, `fra`, and `deu`, then click `Use this pack`. Every Language field on every page and in the graph panel now shows only those three options.

— Example walks through creating and applying a custom pack end to end. `v1.22.0`

---

04 **SKOS validation expanded to 20 checks in three tiers** IMPROVED 85

Expands `validate_skos` from 5 to 20 checks across three tiers — Errors, Warnings, and Info — covering missing `prefLabel`, duplicate language tags, empty labels, `broader` cycles, self-relations, dangling relations, orphans, disconnected components, and more; the two softer tiers are gated so large imports can be checked for errors alone.

— Names the function and every check type but gives no invocation example. `v1.23.0`

---

05 **SKOS autofix for seven check classes** NEW 85

Adds `autofix_skos` to repair seven check classes in one command and report how many instances it resolved: untagged labels, self-relations, dangling relations, top concepts carrying a `broader`, label overlap, redundant hierarchy, and `broader` cycles.

— Names the function and every fixed check class; no example given. v1.23.0

06 **Focus mode seed paste and restore** NEW 80

Adds a paste/restore box to focus mode so focus seeds can be written down and pasted back, matching the node filter's existing restore-a-view syntax; plain names resolve across all focusable kinds, and `Class:Person` `-style` prefixes disambiguate when two kinds share a name.

Restore a saved focus view after returning to a session — paste previously copied seeds including mixed kinds and disambiguated names.

📍 In the graph panel, open Focus mode, click the restore-a-view box, and paste your saved seed list, e.g. Person Class:Person skos:Concept1, then apply.

— Example gives the exact seed syntax to paste and restore. v1.24.0

07 **Delete classes, properties, individuals from graph panel** NEW 80

Adds delete capability to the graph details panel for classes, object and data properties, and individuals (deleted by URI), and for class relations and restrictions (deleted as triples), behind a two-click confirmation modal.

Remove a class or property directly from the graph without navigating away — useful when cleaning up an ontology mid-session.

📍 In the Visualization page, click a class, object property, data property, or individual node to open the details panel, then click Delete and confirm in the modal. To remove a relation or restriction edge, click the edge to open its panel and follow the same two-click flow.

— Example gives the exact two-click delete flow. v1.21.0

08 **Dublin Core metadata on concept schemes** NEW 80

Adds `add_concept_scheme` support for twelve Dublin Core fields — including `dcterms:title`, `creator`, `license`, and dates — written in the correct shapes (typed literals for dates, IRIs for licences and creators).

— Names the function and sample fields but not all twelve. v1.23.0

09 **Annotation type rename across ontology** NEW 80

Adds an Annotation Types tab listing custom annotation types with usage counts and a per-row rename that updates the `owl:AnnotationProperty` declaration, all

annotations using the type as predicate, and any object references; the rename is tracked in undo history.

Rename a custom annotation type that was misspelled at creation time, updating every triple that references it across the ontology in one operation.

📍 Go to Annotations › Annotation Types. Locate the row for the type to rename, click its rename control, type the corrected name, and confirm. The tab shows the updated name and the usage count is unchanged; undo is available if the rename needs to be rolled back.

— Example shows the exact rename flow with undo support noted. v1.22.0

10 **Auto-show and manual reveal for new entities in filtered graphs**

NEW 75

Adds an Auto-show new toggle in the Node options panel so entities created while a filter is active automatically join the current selection, replacing the previous fixed behaviour. Also adds a Show new (N) button beside Select all on the graph page that appends entities hidden by a narrowed node filter to the current selection without replacing the curated view.

When building a small ontology class by class, enable Auto-show new so each entity you create is immediately visible on the canvas without manual filter updates.

📍 In the graph view, open the Node options panel and toggle Auto-show new on.

— Names both controls and their exact scope; example covers only one. v1.25.0 v1.24.0

11 **Visualization page canvas and panel layout improvements**

IMPROVED 75

Fullscreen mode now hides app chrome only — no longer triggering OS fullscreen — keeping Display options, Find and focus, Path finder, Node options, and the details panel reachable. The Node options card now closes automatically when a node or edge is selected. All three Visualization panels (details panel, reopen toggle, Node options picker) float over the canvas as solid cards, preserving full canvas size. Visualization display options collapse behind a persistent toggle, recovering ~100px of canvas height; fit-to-window now reserves the actual space below the graph rather than a flat 90px; page gutters were reduced from Streamlit's 80px to 32px, returning ~110px of width; and ~330px of sidebar spacing was recovered so Quick Stats is no longer pushed below the fold on a 1080p screen.

— Exact pixel gains named but changes are automatic, not user actions. v1.27.2 v1.24.0 v1.23.0 v1.21.0

12 **Full concept editing and relation removal in SKOS page** IMPROVED

75

Supports editing concepts with all three SKOS label kinds (`prefLabel` , `altLabel` , `hiddenLabel`), all seven documentation properties, `notation` , top-concept membership, and cross-vocabulary mappings. Also enables removal of concept relations, which was not previously possible — relations could be added but never taken back, leaving auto-added inverses as half-edges.

— Names every label kind and property type; no UI navigation given. v1.23.0

---

13 **Modifier-click behavior in focus mode** IMPROVED 70

Alt-click in the graph now replaces the entire focus selection with the clicked node, complementing the existing Ctrl/Cmd-click that adds a node. Both are now reversible on their own action: Ctrl/Cmd-click removes an already-focused node (exiting focus mode if it was the last), and Alt-click on the currently focused node exits while preserving seeds for re-entry.

Jump between unrelated nodes in the graph one at a time without having to clear the focus picker between hops.

📍 In the graph panel, Alt-click any node to set it as the sole focus. Alt-click a different node to move focus to that node alone, replacing the previous selection immediately.

— Example demonstrates the exact click sequence and resulting state. v1.25.0 v1.22.0

---

14 **Inline annotation editing via pencil icon** NEW 70

Adds inline editing to each annotation row on the Annotations page via a pencil icon, replacing the delete-and-re-add workflow and preserving language tags and datatypes.

Fix a mislabelled annotation without losing its language tag or datatype — avoids the old delete-and-re-add cycle.

📍 On the Annotations page, locate the annotation row to correct, click the pencil icon to open the inline editor, update the value, and save. The language tag and datatype are preserved automatically.

— Example shows the exact edit flow with tags preserved. v1.21.0

---

15 **Add parent class action in graph panel** NEW 65

Adds an Add parent class action in the graph panel's context actions, inserting a new class above the selected one alongside any existing parents without a detour to the Classes page.

Insert a parent class above an existing node directly from the graph, without leaving to the Classes page and manually re-parenting.

📍 In the graph panel, select a class node, then choose Add parent class from the context actions. The new class is added alongside any existing parents of the selected class.

— Example gives exact steps but no deeper mechanism beyond the action. v1.27.3

16 **Clear button on entity dropdowns** NEW 65

Adds a clear cross to every entity dropdown — annotation types, namespaces, domains and ranges, class and property pickers, relation endpoints, restriction values, SKOS schemes and broader concepts, and plain view selectors — with a named validation error when a required field is cleared.

— Every dropdown type named but no exact UI path given. v1.21.0

17 **SKOS-XL label support** NEW 60

Adds SKOS-XL support so vocabularies like AGROVOC and EuroVoc that publish labels as resources rather than literals arrive correctly labelled instead of reporting `missing_prefLabel` for every concept.

— Names affected vocabularies and error avoided but no usage steps. v1.23.0

18 **Python 3.12 minimum version** BREAKING 60

`requires-python` floor raised from 3.10 to 3.12; environments running Python 3.10 or 3.11 will no longer work.

— Exact version requirement given; clear upgrade trigger. v1.23.0

THINNER COVERAGE BELOW

19 **Copy buttons for canvas label and status bar** NEW 55

Adds a copy button on the canvas for the selected node's label and a copy button at the end of the status bar for the full status-bar line, giving access to annotation values previously truncated in place.

Copy a long annotation value that the status bar has ellipsised, so you can paste the full string into another tool.

📍 Select the annotated node on the canvas, then click the copy button at the end of the status bar to copy the full status-bar line to the clipboard.

— Clear UI location, limited mechanism beyond the copy action. v1.24.0

20 **Explicit notices for focus and filter edge cases** IMPROVED 50

Focus mode now ends with an on-screen notice instead of silently substituting an arbitrary class when no seeds remain or the focused entity's type is switched off. The Visualization page shows a visible notice when nodes are hidden due to an active focus mode or narrowed filter. A focus with insufficient room for its annotations now reports the problem explicitly instead of silently drawing nothing when Annotations is ticked.

– Names three trigger conditions but not exact wording or resolution steps. v1.27.3 v1.25.0 v1.21.0

21 **pyvis dependency removed** BREAKING 50

`pyvis` is no longer installed as a dependency; anything relying on it arriving transitively from OrionBelt must now declare it explicitly.

– States the exact package and required action, nothing more. v1.25.0

22 **subClassOf cycle detection warnings** NEW 45

Reports `subClassOf` cycles as warnings when a class points to one of its own descendants, surfacing loops that previously went unmentioned by validation.

– Describes the trigger but not where the warning surfaces or how to fix it. v1.27.3

23 **Undo/Redo preserves active node filters** IMPROVED 35

Undo and Redo now preserve the graph's active node filters, so narrowing the class filter and undoing no longer floods the canvas with every class.

– Bug-fix description only, nothing for a reader to act on. v1.27.3

┗──▷ BREAKING ON UPGRADE

! `pyvis` is no longer installed as a dependency; anything relying on it arriving transitively from OrionBelt must now declare it explicitly.

! Focus mode can now end by itself when there is nothing left to focus on, where it previously substituted a class.

! A modifier-click (Ctrl/Cmd or Alt) on an already-focused node now reverses the focus action, where it previously did nothing.

! A class created while the node filter is narrowed no longer appears in the graph automatically — it is held back and surfaced via the Show new (N) button and a toast notification instead.

! Python 3.12 is now the minimum required version (`requires-python` floor raised from 3.10 to 3.12); environments running Python 3.10 or 3.11 will no longer work.

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ 1 RELEASE · 2026-08-10

NOTES ↗ RANK

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

> LangSmith adds a `POST /runs/rules/validate` endpoint for testing thread evaluators, enforces monthly trace limits scoped to projects and users, surfaces OpenTelemetry resource attributes as trace metadata via `OTEL_RESOURCE_ATTRIBUTES`, and switches bulk exports to zstd compression by default — while removing legacy dataset comparison SDK helpers in favor of a new experiment-runs endpoint.

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

01 Thread evaluator validation endpoint NEW

90

New `POST /runs/rules/validate` endpoint lets you test a multi-turn thread evaluator against a real conversation, passing `test_thread_id` and `session_id`, before saving it — catching mapping errors without affecting production runs.

Test a multi-turn thread evaluator against a real conversation before saving it, to catch mapping errors without affecting production runs.

```
$ curl -X POST 'https://<langsmith-host>/runs/rules/validate' \
  -H 'Content-Type: application/json' \
  -H 'X-API-Key: <your-api-key>' \
  -d '{"test_thread_id": "<thread-uuid>", "session_id": "
<session-uuid>"}'
```

— Runnable curl example against a named endpoint with parameters. snapshot-20260908

02 OpenTelemetry resource attributes as trace metadata NEW

90

LangSmith now reads `OTEL_RESOURCE_ATTRIBUTES` so OpenTelemetry resource attributes appear on traces as metadata namespaced under `otel.resource.*`, letting teams attach details like user IDs without changing span emission code; filter by `otel.resource.*` fields in the LangSmith UI.

Attach deployment metadata (service name, user ID) to every trace without changing how your tracer emits spans, then filter by `otel.resource.*` fields in the LangSmith UI.

```
$ export OTEL_RESOURCE_ATTRIBUTES="service.name=my-agent,user.id=u_12345"
```

— Exact env var, namespace, and runnable export command given. snapshot-20260908

03 **Legacy dataset comparison SDK helpers removed** BREAKING

80

Legacy dataset comparison helpers are removed from the public OpenAPI spec and generated SDKs; code using those SDK methods must migrate to `POST /v2/datasets/{dataset_id}/experiment-runs`.

— Names removed helpers and the exact replacement endpoint. snapshot-20260908

04 **Zstd compression default for bulk exports** IMPROVED

75

Bulk export of experiments and tracing data now defaults to zstd compression; self-hosted deployments can keep gzip compatibility with existing downstream consumers by setting `FF_BULK_EXPORT_DEFAULT_COMPRESSION=gzip` in `.env`.

Keep gzip compression for bulk exports on a self-hosted deployment when you need compatibility with existing downstream consumers.

```
$ FF_BULK_EXPORT_DEFAULT_COMPRESSION=gzip
```

— Named config key and file for controlling the new default. snapshot-20260908

THINNER COVERAGE BELOW

05 **Bulk dataset split management in experiment tables** NEW

55

Select multiple rows in experiment tables — or all rows matching current filters — to add, replace, or remove dataset splits in one action, or copy selected examples to another dataset.

— Clear UI workflow but no exact navigation or command. snapshot-20260908

06 **Monthly trace limits per project and user** NEW

50

Enforces user-defined monthly trace limits scoped to individual projects and users, rejecting new traces that exceed a configured limit while still allowing patches and feedback on already-accepted traces.

— Explains behavior but no config surface or UI path given. snapshot-20260908

07 **Error visibility for failing custom code evaluators** IMPROVED

35

Custom code evaluators that time out or fail now record an error on that run instead of silently leaving it without feedback, making partial evaluation failures visible in experiments.

— Describes behavior change with no named surface or steps. snapshot-20260908

---

08 **Clearer error for oversized CSV exports** IMPROVED 35

Exporting a dataset comparison view as CSV now returns a 'file is too large to export' error instead of a generic server error when the export exceeds internal size limits.

— Names the error message but no threshold or config. snapshot-20260908

---

09 **Thread evaluator config preview accuracy** IMPROVED 30

Thread evaluator config preview now shows only the thread message formats the evaluator actually maps, instead of listing every available format.

— Thin UI-only description with no mechanism detail. snapshot-20260908

└──▷ BREAKING ON UPGRADE

! Legacy dataset comparison helpers are removed from the public OpenAPI spec and generated SDKs; code using those SDK methods must migrate to `POST /v2/datasets/{dataset_id}/experiment-runs`.

WAS THIS USEFUL? 👍 👎

Comet Opik

Feedback ⓘ 2 RELEASES · 2026-09-07 → 2026-09-08 NOTES ↗

RANK

Opik traces, evaluates, and monitors LLM applications and agentic workflows, with datasets, an LLM-as-judge metric library, and production dashboards.

> Opik added Cerebras support to its Python SDK for tracing LLM calls, and improved MCP OAuth token introspection to report expiration.

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

01 **MCP OAuth token introspection expiry reporting** IMPROVED

50

MCP OAuth token introspection responses now report `expires_at`.

— Names exact field but no usage context 2.2.53

02 **Cerebras integration in Python SDK** NEW

40

Adds Cerebras integration to the Opik Python SDK for tracing and observability of Cerebras-hosted LLM calls.

— Names integration but no setup detail or mechanism 2.2.54

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ 1 RELEASE · 2026-09-01 NOTES ↗ RANK

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

> Braintrust moved Loop to a persistent, Braintrust-managed server-side runtime and added Patterns scheduled investigations and Loop automations on top of it, added the glm-5.3-flash model to its Gateway, and shipped new tracing/eval instrumentation across the TypeScript SDK, Python SDK, GitHub eval action, and Harbor plugin.

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

01 Score/metric filtering in GitHub eval action NEW 90

Adds `report_scores` and `report_metrics` inputs to the GitHub eval action (v2.1.0) to filter which scores and metrics appear in the PR comment, each accepting a comma- or newline-separated list of names.

Restrict a PR comment to only the scores and metrics your team cares about, keeping the table focused during code review.

yaml

```yaml
- uses: braintrustdata/eval-action@v2
  with:
    report_scores: accuracy,faithfulness
    report_metrics: latency,cost
```

— Named config keys with a runnable workflow example snapshot-20260908

02 Agno eval instrumentation in Python SDK NEW 80

Adds `AccuracyEval`, `AgentAsJudgeEval`, `ReliabilityEval`, and `PerformanceEval` eval instrumentation for Agno in the Python SDK (v0.36.0), with eval suites automatically traced via auto_instrument() and experiments opened in Braintrust automatically.

— Names classes and the auto_instrument function, no full example snapshot-20260908

03 Loop moved to managed server-side runtime IMPROVED 75

Moves Loop to a Braintrust-managed server-side runtime so investigations persist across sessions, operate across logs, experiments, and datasets, and can create or edit prompts, scorers, datasets, facets, custom views, dashboards, and automations — with every write pausing for approval unless auto-accept is enabled.

— Explains new persistence and write scope, but no UI path or command shown

snapshot-20260908

04 **Verifier output upload in Harbor plugin** NEW 70

Adds standard verifier output upload to the Harbor plugin in the Python SDK (v0.36.0): `test-stdout.txt`, `test-stderr.txt`, and `ctrf.json` with `attachments` and `redact_patterns` configuration.

— Names files and config keys but gives no configuration example snapshot-20260908

05 **Loop automations for scheduled runs** NEW 65

Adds Loop automations: scheduled Loop runs with their own instruction, model, and write permissions that leave a read-only thread you can Continue, with results optionally sent to a Slack channel or webhook.

— Names scheduling, permissions, and delivery targets, no setup steps snapshot-20260908

THINNER COVERAGE BELOW

06 **glm-5.3-flash model in Braintrust Gateway** NEW 55

Adds `glm-5.3-flash` as a built-in multimodal reasoning model available via the Braintrust Gateway and selectable under the Braintrust provider in playgrounds, prompts, and scorers — no AI provider setup required.

— Names surfaces where model is selectable but no exact steps shown snapshot-20260908

07 **Column name conflict validation** IMPROVED 30

Custom column names that conflict with built-in table fields are now rejected at creation time.

— Bare statement of new validation with no scope or example snapshot-20260908

WAS THIS USEFUL? 👍 👎

Langfuse

Feedback ⓘ 2 RELEASES · 2026-09-08 NOTES ↗ RANK

Langfuse provides tracing, evaluation, and monitoring for LLM applications.

> Langfuse shipped new operational alerting and export-lag metrics alongside UI improvements to message rendering and cost breakdowns, while removing unstable evaluation API endpoints in a breaking change.

└──▷ **WHAT SHIPPED · 5 FEATURES** most completely described first ? what's the number?

01 **Improved message rendering preview** NEW 50

Introduces an Improved Message Rendering preview feature with a normalized I/O parser for trace inputs and outputs.

– Named preview feature but no toggle path or flag specified. v4.31.0

02 **Removal of unstable evaluation API endpoints** BREAKING 45

Unstable evaluation API endpoints that were previously available but not production-ready have been removed; any integrations calling those endpoints will break on upgrade.

– Clear breaking change and migration note, but no specific endpoint paths named. v4.31.0

03 **Export data-freshness lag metrics** NEW 43

The worker now emits export data-freshness lag distribution metrics, enabling monitoring of how stale exported data is.

– Names metric type but no metric name or endpoint given. v4.32.0

04 **Alerting for unhealthy labeled previews** NEW 35

Langfuse now alerts when a labeled preview is not serving, surfacing unhealthy preview deployments so teams can detect deployment issues early.

– Describes trigger but no config or endpoint given. v4.32.0

05 **Cost source in trace cost tooltip** IMPROVED 35

The trace cost breakdown tooltip now shows the cost source.

– Clear UI location to view the change, but minimal mechanism described. v4.31.0

└──▷ **BREAKING ON UPGRADE**

! The unstable evaluation API endpoints have been removed; any integrations calling those endpoints will break on upgrade.

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

Certiv

Feedback ⓘ 1 RELEASE · 2026-09-04 NOTES ↗ RANK

Certiv provides endpoint visibility and control for locally installed AI agents, helping organizations discover, understand, and safeguard AI actions.

> Certiv Cost enters public preview, giving teams per-endpoint token budgets with enforcement controls, session-level spend attribution, and a user-facing spend view in Scout.

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

01 Per-endpoint token budgets with enforcement modes and simulation

NEW 93

Certiv Cost adds per-endpoint dollar budgets with three enforcement modes — Report, Notify, and Block — where Block stops prompts at the endpoint before the request is sent. Budgets support custom contract rates entered per model so thresholds reflect what the organization actually pays rather than list pricing, and budget simulation uses historical usage to preview how often an endpoint would have hit a proposed threshold before it is enforced.

Start monitoring token spend without interrupting work — switch to Block only after you understand normal consumption patterns for an endpoint.

> 📍 In the Certiv console, go to the endpoint you want to protect, open Cost settings, set a dollar budget for your chosen period, and set the enforcement mode to Notify. Once you have a clear picture of historical usage, use the budget simulation to pick a realistic ceiling, then switch enforcement to Block.

— Names all three enforcement modes plus config steps to apply them

Token Spend Is a Potential Signal for a…

THINNER COVERAGE BELOW

02 Session-level token attribution tracking NEW

48

Certiv Cost adds session-level token attribution, tracking spend by agent session, endpoint, user, and model with historical trend data.

— Describes scope of tracking but no UI or config path given

Token Spend Is a Potential Signal for a…

03 Scout end-user spend view NEW

40

Adds a Scout end-user spend view showing each user their own token consumption on their own endpoint, broken out by hour.

— States what the view shows but no navigation detail

Token Spend Is a Potential Signal for a…

WAS THIS USEFUL?  

ToolHive

Feedback ⓘ 1 RELEASE · 2026-09-08 NOTES ↗ RANK

ToolHive runs MCP servers in isolated containers and enforces identity and access policies for local and Kubernetes deployments.

> ToolHive v0.47.0 focuses on trust and verification: SPIFFE-based identity for operator CRDs, cosign key-signed verification for skills and plugins, and several OAuth/OIDC hardening and delegation changes.

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

01 Key-signed skill and plugin verification via cosign NEW

70

Verifies key-signed skill and plugin installs against a public key, recording the pinned cosign key in the lock schema. Honours the pinned cosign key on sync and upgrade, signs plugin pushes with the lock feature gate removed, and reports key-signed artifacts as such at install time.

— Names cosign, lock schema and lifecycle stages affected v0.47.0

THINNER COVERAGE BELOW

02 OIDC and OAuth callback security hardening IMPROVED

45

Guards OIDC discovery against private IPs and suppresses the OAuth callback Referer header.

— Two named security fixes, no mechanism detail v0.47.0

03 SPIFFE trust configuration in operator CRDs NEW

43

Defines SPIFFE trust configuration and exposes SPIFFE client-auth registration through operator CRDs.

— Names SPIFFE and CRDs but no mechanism detail v0.47.0

04 Private CA trust for RFC 8693 trusted issuers IMPROVED

40

Trusts private CAs for RFC 8693 trusted issuers.

— Names RFC 8693 but no config detail v0.47.0

05 LLM gateway and VS Code integration updates IMPROVED

35

Routes Claude workflows through the Stacklok LLM gateway and updates VS Code LLM integration.

— Names integrations but lacks specifics of the update v0.47.0

06 **Configurable proxy request-body limit** IMPROVED 30

Makes the proxy request-body limit configurable.

— Bare description, no key or default value given v0.47.0

07 **Cedar delegation without upstream session** IMPROVED 30

Allows Cedar delegation without an upstream session.

— Names Cedar delegation but no mechanism v0.47.0

08 **Canonical inbound grants in CRDs** NEW 30

Exposes canonical inbound grants in CRDs.

— Bare mention of CRD field, no detail v0.47.0

09 **Embedded auth servers without upstreams** NEW 30

Allows embedded auth servers without upstreams.

— States capability without operational detail v0.47.0

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · September 8, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe | Feedback on this issue