THE AI TOOLCHAIN

NO. 079

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED OCTOBER 31, 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 345 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 75 of 77 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. Pinecone's pg_pinecone puts filtered vector search, embedding generation and BM25 inside Postgres, so a stack can stop syncing a second store. Weaviate 1.40 makes 4-bit

vim ? help

Rotational Quantization GA, cutting vector memory. And fal plus PydanticAI both added fallback chains, so one dead model no longer fails the request.

BUILD

Run retrieval where your relational data already lives

pg_pinecone adds filtered vector search, in-database embedding generation, BM25 full-text and reranking as a Postgres extension — the join between metadata and vectors stops being an application-layer merge across two systems. Weaviate 1.40's GA 4-bit Rotational Quantization and MUVERA multi-vector indexing cut the memory bill for keeping those vectors hot, and a named vector index can now be dropped without rebuilding the collection.

Pinecone · Weaviate

DEPLOY

Survive a bad model or provider without writing the retry logic

fal's fallback chains let one API request try up to three models in sequence when earlier ones fail; PydanticAI's ModelFallback adds sticky routing and stream-timeout triggers, so a response that stalls mid-stream reroutes instead of hanging. Both remove the hand-rolled try/except ladder that every agent codebase ends up carrying.

fal · PydanticAI

GOVERN

Cap spend and block prompt attacks at the org boundary, not per repo

Claude Code's gateway admin surface adds layered policy merging, Bedrock guardrail prompt-attack filtering and pooled group spend budgets — a team's overrun is visible as a budget rather than discovered on the invoice. magpie's scoped short-lived gateway keys mean a leaked credential expires on its own instead of needing a manual rotation sweep.

Claude Code · magpie

BUILD

◆ AI Coding Agents

yetone magpie

Feedback ⓘ v0.1.2359 CODE ↗ RANK

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

> Magpie shipped scoped short-lived gateway keys, per-provider plugin option persistence, and a faster usage-log/requests engine, alongside smaller additions like custom local model IDs, a `magpie log` command, install.sh uninstall/purge flags, subagent syncing, and clearer provider failure messages.

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

01 **Short-lived scoped gateway keys** NEW 95

Issue short-lived gateway keys scoped to specific models via `POST /v1/magpie/keys` or the `magpie gateway-key issue` command, with a TTL between 1 minute and 30 days controlled by `MinTTL` / `MaxTTL`. Issued keys are bounded by their issuer key: they are refused if the issuer is removed, disabled, or expired, inherit the issuer's allowed models, and count against the issuer's budget limits.

— Exact endpoint and CLI command with full TTL and inheritance mechanism given v0.1.2334

02 **`magpie log` command for the desktop app log** NEW 85

Access the desktop app's log file with `magpie log [--path] [-n N] [-f]`, which shows the path to `app.log`, tails it, or follows it live — covering the app, menu bar, and panel regardless of how they were opened.

Follow the desktop app log live to diagnose a gateway issue without opening a terminal at launch.

```
$ magpie log -f
```

Get the exact path to `app.log` to attach it to a bug report.

```
$ magpie log --path
```

— Exact flags with runnable example commands for tailing and locating the log v0.1.2342

03 **install.sh `--uninstall` and `--purge` flags** NEW 80

`--uninstall` removes the magpie binary, app, and login service while keeping settings, providers and usage data in `~/.config/magpie` for a future reinstall; `--purge` fully removes magpie, disconnects all agents, and deletes all settings, providers and usage data.

Remove magpie cleanly but keep your settings so you can reinstall later without reconfiguring.

```
$ curl -fsSL https://usemagpie.ai/install.sh | sh -s -- --uninstall
```

Fully remove magpie, disconnect all agents, and delete all settings, providers and usage data.

```
$ curl -fsSL https://usemagpie.ai/install.sh | sh -s -- --purge
```

— Exact flags with runnable curl|sh commands and clear behavioral difference v0.1.2354

04 **Per-provider plugin options persistence** NEW 70

The `magpie plugin options <name>` command now reads and writes the switches a provider plugin declares (its editor's Plugin options), persisted per-provider via the new `EditPluginOptions` function, in addition to the existing middleware `ctx.options` path.

— Names command and internal function but no example invocation v0.1.2359

05 **Usage log and requests engine performance improvements**

IMPROVED 70

The usage log now keeps an incremental index of immutable compressed log blocks, reusing sealed blocks for Summary, Vias, and request pages instead of re-decoding the full JSONL history, cutting a 100k-record append + All-page benchmark from 119.5 ms / 117.6 MB to 88.5 ms / 67.1 MB. The request page also caches priced gateway blocks within its existing 24 MiB source-cache budget, invalidating on price overrides, live catalog changes, provider identity changes, or renames, and CSV fallback matching now indexes request IDs and token/session/time candidates instead of comparing every session call against every gateway call.

— Detailed mechanism and benchmarks, but nothing a user directly operates v0.1.2348 v0.1.2359

06 **Vendor error messages on resting provider rows** IMPROVED 70

Resting provider rows now show the vendor's own error message (via the `error` field, up to 300 runes, in the `Rest` struct) alongside the failure status, so a 'failed · back in 16s' entry explains why — e.g. 'model not available for your plan'. The routing UI also displays the HTTP status code prefix and a tooltip with the vendor's words on plain `other` failures via the `restSay` and `restTip` functions in `client.js`.

— Named struct field, limit, and functions, but purely informational display v0.1.2340

07 **Custom local model IDs via `ModelIDs`** NEW 65

List a provider's model under a user-chosen ID via `ModelIDs` in settings, while requests go out to the provider under its own ID — letting the same model appear under different names across relays, e.g. `deepseek-v4.1-flash-1017` or `grok-4.6@500k`.

— Named config key and example IDs, but no file path or command shown v0.1.2345

08 **Subagents included in sync and backup** IMPROVED 65

Sync and backup files now carry subagents — a subagent saved on one computer is written into the other's agent directories (`.claude/agents/` and `.codex/agents/`) on the next sync, and removed when taken out on the source computer.

— Named directories and propagation/removal behavior, but no manual trigger shown v0.1.2333

THINNER COVERAGE BELOW

09 **Model filter on Requests usage page** NEW 55

The Requests usage page now has a Model filter (`ledModel`) beside the existing Provider filter, scoping all page totals, cells, and charts to a selected model's token usage.

— Named field and clear UI location, no further mechanism given v0.1.2358

10 **Masking alert toggle on Masking page** IMPROVED 55

Adds a `Masking alert` toggle (`maskAlert` preference, On/Off) to the Masking page, so the notification shown when a value is masked for the first time is now configured there instead of in Settings › Usage.

— Named preference key and exact UI relocation, no further detail v0.1.2336

NEW SUPPORTED PLATFORM ChatGPT desktop app (codex) on Linux

v0.1.2353 CODE ↗ STABLE

magpie now detects and switches accounts for the ChatGPT desktop app on Linux, reading version metadata from `resources/linux-package-metadata.json`.

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.2353
```

git

```
$ git clone --branch v0.1.2353 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.2353
```

published under the project's GitHub account · checked 31 Oct Wrong command?

Detect the ChatGPT desktop app's codex process on Linux by recognising executables under `/usr/lib/chatgpt/resources/codex`, matching the parity macOS and Windows already had via linuxCodexApp(exe) inside `isCodexAppServer`.

Adds a `linux` field to the `desktopApp` struct, set to `/usr/lib/chatgpt` for the `codex` entry, so magpie can locate the app installed by OpenAI's chatgpt deb, rpm, or Arch package (preview since 2024-08-11) and read its version from `resources/linux-package-metadata.json`.

WAS THIS USEFUL? 👍 👎

Yuval Sarel cones

Feedback ⓘ v0.4.8 NOTES ↗ RANK

Cones is a terminal dashboard for managing coding-agent sessions across projects.

> cones v0.4.8 adds nested session parenting, full OpenCode 2 support, and fixes a long-instruction launch failure for pi agents.

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

01 **Nested session parenting via --parent** NEW 86

New sessions can be placed under a specific parent or the current session using `cones launch --parent ID` or `cones launch --parent self`; a non-live parent is treated as an error. Sessions started inside another session (for example, `opencode run` from an agent's shell) are now automatically listed as children of the parent session in the dashboard.

— Names exact flag, values, and error behaviour v0.4.8

02 **OpenCode 2 support** NEW 62

cones now supports OpenCode 2 (verified on 2.2.0), including live columns, notes, side questions, history, and resume. It also restores the `--model` flag, which OpenCode 2 removed from its own interactive client.

— Names version and flag but mechanism of each feature is thin v0.4.8

THINNER COVERAGE BELOW

03 **Long-instruction handling for pi agents** IMPROVED 53

Instructions over 512 bytes now reach pi agents outside the command line, removing a launch failure that previously occurred with long instructions.

— Gives exact size threshold but no reader action v0.4.8

WAS THIS USEFUL? 👍 👎

octomind

Feedback ⓘ 0.67.0 NOTES ↗ RANK

Open-source AI coding agent and agent runtime: one binary, any model, MCP-native. Runs in terminal, CI, or as a daemon.

> octomind 0.67.0 reworks request transport to be configurable per model, adds scheduled MCP tool calls and per-session cost tracking, and lets specialists queue follow-up prompts via `tap`.

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

01 Queued follow-up prompts via `tap` NEW

65

Follow-up prompts can be queued for a specialist that is already running via `tap`, so messages are delivered in order without waiting for the current turn to finish.

– Names the `tap` mechanism, a concrete starting point 0.67.0

02 Per-model transport configuration (WebSocket/HTTP) BREAKING

60

Request transport (WebSocket or HTTP) is now configured per model in the config rather than globally or per provider. ChatGPT providers now use WebSockets for communication, matching the configurable transport behavior of the octohub provider. This is a breaking change: existing configurations that assume a global or provider-level transport setting may need to be updated to the new per-model format.

– Mechanism and providers named, but no exact config key shown 0.67.0

THINNER COVERAGE BELOW

03 Per-model and per-role stream inactivity timeouts NEW

45

Separate stream inactivity timeouts can now be set per model or per role, controlling how long octomind waits before treating a stalled stream as failed.

– Behavior explained but no config key or flag given 0.67.0

04 Scheduled MCP tool calls NEW

45

MCP tool calls can now be scheduled to run at a defined time, enabling deferred or cron-style agent actions without an active interactive session.

– Describes capability but no scheduling syntax or command shown 0.67.0

05 Agent run history with cost tracking NEW

45

octomind now tracks agent run history and main-model costs per session, surfacing spend alongside task history.

— States what's tracked but not how to view it 0.67.0

└──▷ **BREAKING ON UPGRADE**

! Request transport is now configured per model in the config; existing configurations that assume a global or provider-level transport setting may need to be updated to the new per-model format.

WAS THIS USEFUL? 👍 👎

keploy

Feedback ⓘ v3.6.184 NOTES ↗ RANK

Open-source platform for creating safe, isolated production sandboxes for API, integration, and E2E testing.

> Keploy's learn mode now supports chaining test values across a test set, letting dependent tests follow a value produced by an earlier test.

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

01 **Shared variables chained across test sets** NEW 40

In learn mode, a test set can now follow a value produced by an earlier test as a shared variable, enabling dependent tests to chain outputs across recordings.

— Describes behaviour but no concrete config key or command shown v3.6.184

WAS THIS USEFUL? 👍 👎

Anthropic Claude Code

Feedback ⓘ v2.1.316 NOTES ↗

CODE ↗ RANK

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

> Claude Code's v2.1.316 release adds several Claude apps gateway admin capabilities — layered policy merging, Bedrock guardrail prompt-attack filtering, and pooled group spend budgets — alongside new environment variables for controlling subagent effort and the ultrathink keyword, a new Code Review severity analytics card, and smaller cloud session UX fixes, while flagging the deprecation of Opus 4.8 fast mode.

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

01 Claude apps gateway policy, guardrail and budget controls

NEW 90

The Claude apps gateway gains `managed.policy_mode: layered`, which applies and merges every matching developer policy in order instead of stopping at the first match; `guardrail.input_tagging` on Bedrock upstreams to enable Amazon Bedrock guardrail prompt-attack filtering, with an `all_messages_and_system` option to also tag the system prompt; a `pooled_budget` field on the spend limits admin API giving a group's members one shared monthly budget on top of each member's own individual limit; and the ability to sign in to a Claude apps gateway directly from the VSCode chat panel with no terminal needed.

Wrap prompts in Amazon Bedrock guardrail input tags to enable prompt-attack filtering on a gateway Bedrock upstream, including the system prompt.

yaml

```yaml
guardrail:
  input_tagging: all_messages_and_system
```

Apply all matching developer policies in a Claude apps gateway rather than stopping at the first match, so layered org-wide and team-specific policies all take effect.

yaml

```yaml
managed:
  policy_mode: layered
```

— Names exact config keys and includes runnable YAML snippets for two of four v2.1.316

02 **Subagent effort environment variable** NEW 70

Set a default effort level for all subagents with the `CLAUDE_CODE_SUBAGENT_EFFORT` environment variable, instead of having subagents inherit the session's effort level — useful for pinning many parallel sub-tasks to a consistent, lower-cost effort.

Prevent subagents from inheriting the session's effort level and instead pin them to a specific effort, such as when running many parallel sub-tasks where you want consistent, lower-cost effort.

```
$ CLAUDE_CODE_SUBAGENT_EFFORT=low claude
```

— Named env var with a runnable example command v2.1.316

03 **Opus 4.8 fast mode deprecation** DEPRECATED 70

Fast mode for Opus 4.8 is deprecated as of October 30, 2026 and will be removed November 30, 2026; after removal, fast mode requests on Opus 4.8 return an error with no fallback to standard Opus 4.8 — migrate to Opus 5.5 to retain the speedup.

— Exact dates and migration path but no code change needed v2.1.316

THINNER COVERAGE BELOW

04 **Ultrathink keyword disable switch** NEW 55

Disable the `ultrathink` keyword within a session by setting the `CLAUDE_CODE_DISABLE_ULTRATHINK` environment variable.

— Named env var but no example or further mechanism given v2.1.316

05 **Code Review findings-by-severity analytics card** NEW 55

The Code Review analytics page now shows a findings-by-severity card with total findings for a period, a daily chart stacked by severity, and counts per severity.

— Describes card contents but only a UI path, no API or config v2.1.316

06 **Cloud session UX improvements** IMPROVED 45

Cloud sessions gained a form card to fill in or decline an MCP server request instead of the request being canceled unseen, and a Catching up… status line shown in cloud and Remote Control sessions while messages from a session you switch back to are still loading.

— Two UI behaviour tweaks with no named config or exact path v2.1.316

07 **Looped cross-certificate preservation** NEW 45

Preserve looped cross-certificates from a private CA by setting `CLAUDE_CODE_DISABLE_LOOPED_CROSS_CERT_DROP`.

— Bare env-var name with no explanation of underlying behaviour v2.1.316

└──▷ **BREAKING ON UPGRADE**

! Fast mode for Opus 4.8 is deprecated as of October 30, 2026 and will be removed November 30, 2026; after removal, fast mode requests on Opus 4.8 return an error and do not fall back to standard Opus 4.8 — migrate to Opus 5.5 to retain the speedup.

WAS THIS USEFUL? 👍 👎

The Open Engine Zeroshot

Feedback ⓘ v10.24.0 NOTES ↗

RANK

Zeroshot orchestrates executor-verifier agent loops that implement and validate software changes while logging workflow state to SQLite.

> Zeroshot's latest releases focus on hosted-run reliability and inspection tooling: shared lesson memory now reconciles across concurrent runs on a new host-owned storage backend, the CLI gained a full template graph inspector, target discovery picked up GPU machine sizes, and the contributed adapter registry expanded with new harnesses and provider lanes.

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

01 **Template graph tree view via `zeroshot template show`**

IMPROVED 95

`zeroshot template show` now renders a tree of every node kind, guard, attempt count, signal, bound, and executable binding marker for a template graph; pass `--json` or pipe the output to get the full GraphSpec JSON.

Dump a template's complete GraphSpec as JSON for programmatic inspection or diffing between versions.

```
$ zeroshot template show software-change --json
```

— Exact command, flag, and runnable example given v10.23.0

02 **Contributed adapter registry: OpenCode, Hermes, Copilot BYOK, Pi GitHub**

NEW 65

Zeroshot now includes contributed OpenCode and Hermes adapters in local builds, sharing the same execution, session, permission, and Active Memory contracts as maintained harnesses across all six existing provider families. Copilot gains native BYOK provider lanes and Pi gains GitHub support, both through the new contributed adapter registry.

— Names adapters and providers but no setup steps v10.22.0

03 **XL and T4 GPU machine sizes in target discovery** NEW

60

Zeroshot now supports XL and T4 GPU machine sizes in target discovery, with the shared size picker showing the box label, lane count, and full Boost cost for the selected

machine.

— Names sizes and picker fields but no exact command v10.23.0

THINNER COVERAGE BELOW

04 **Shared lesson memory reconciliation for hosted runs** NEW

50

When a concurrent hosted run updates a lesson, Zeroshot refreshes shared memory before postmortem and presents the maintainer with the run's original advice alongside the current version, keeping shared lesson memory consistent across concurrent runs.

— Mechanism explained but no command or config surface given v10.24.0

05 **Host-owned async storage backend for hosted run memory** NEW

45

Zeroshot adds a host-owned asynchronous storage backend with scoped reads and atomic publications so hosting adapters can authenticate memory access for hosted runs.

— Architecture described but no reader-facing action v10.24.0

WAS THIS USEFUL?  

Graphify Labs Graphify

Feedback ⓘ v0.9.99 NOTES ↗

RANK

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

> Graphify's v0.9.98 and v0.9.99 releases focus entirely on confidence scoring for graph traversal, exposing how certain each relationship and path really is.

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

01 Confidence scoring for paths, explain, and neighbors NEW

84

Graphify's `graphify path` command now reports overall path confidence as the weakest hop (lowest confidence tier, then lowest score), with a summary line naming that hop. When multiple shortest paths tie on length, `graphify path` and the MCP `shortest_path` tool now pick the most conservative route — the one whose weakest hop is weakest — and note it was the weakest of N equal-length routes. `serve` output for `path`, `explain`, and `neighbors` / `get_neighbors` now prints each edge's stored `confidence_score` alongside its tag, e.g. `[INFERRED 0.75]`; `EXTRACTED` edges stay bare (1.0 by definition), and missing or non-finite scores also keep the bare tag.

Trace a call route between two members and see which hop is the weakest confidence link — useful when auditing inferred edges in a code graph.

```
$ graphify path "Exporter.Run" "PdfReport.Build"
```

Trace the most conservative path between two concepts when multiple equal-length routes exist, to avoid overstating connection confidence.

```
$ graphify path "FastAPI" "ModelField"
```

Inspect edge confidence scores while exploring a concept, to distinguish strongly extracted links from weakly inferred ones.

```
$ graphify explain "APIRouter"
```

— Names exact commands, tags, and scoring rules; clear how to invoke. v0.9.98 v0.9.99

WAS THIS USEFUL?  

forgedock

Feedback ⓘ v1.38.0 NOTES ↗ RANK

ForgeDock — autonomous AI dev pipeline for Claude Code. GitHub as agent memory: issue in, PR out.

> ForgeDock's v1.38.0 release adds autonomous decision-making, issue dependency tracking, configurable issue watches with threshold-triggered releases, production-signal adapters, and a merge gate for hand-built PRs, alongside internal scheduling and capability-pinning improvements.

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

01 **Named scoped issue watches** NEW 80

ForgeDock adds named, scoped, capped issue watches with milestone, version, label, and search selectors, plus `max_issues` and `until` caps, a forever mode, and an optional long-lived batch.

— Lists concrete selectors and caps but no command syntax v1.38.0

02 **North-star decision engine for escalations** NEW 70

ForgeDock now settles decision-only escalations automatically using a `NORTHSTAR` file and anchored decisions in the target repo; external actions still require a human.

— Names the NORTHSTAR file and scope but no exact usage steps v1.38.0

03 **Issue-level dependency and hold tracking** NEW 70

ForgeDock adds issue-level dependency and hold tracking, parsing `Blocked by #N` lines into DAG edges and honouring a blocked label on `--add`.

— Names the flag and label mechanism clearly v1.38.0

04 **Threshold-triggered themed releases** NEW 70

ForgeDock adds threshold-triggered themed releases in issue-watch mode — pools issues per target, cuts a security, stealth, or similarly themed release when a threshold is crossed, orchestrates it, and opens the deploy PR.

— Explains trigger and output but no configuration detail v1.38.0

05 **Production signal adapters for watches** NEW 70

ForgeDock adds a signal adapter contract for production watches, with memo, Alertmanager, SQL, and Prometheus adapters, all enforced read-only.

— Names all four adapters but no setup steps v1.38.0

06 **Merge gate for hand-built PRs** NEW 65

ForgeDock adds a merge-gate mode for hand-built PRs, blocking owner-approved PRs that lack a `/work-on` phase trail from merging through the gate.

— Names the phase-trail requirement but not how to enable the gate v1.38.0

THINNER COVERAGE BELOW

07 **Default-off rollout lifecycle support** NEW 55

ForgeDock adds default-off rollout lifecycle support — rollouts ship dormant with no activation owner, plus an activation row, shadow-mode conformance check, owner ask, and dormancy report.

— Describes lifecycle stages but no concrete config or command v1.38.0

08 **Continuous local lane concurrency scaling** IMPROVED 40

ForgeDock now scales local lane concurrency continuously from available cores and memory, and runs validation through the heavy semaphore.

— Explains behaviour change but no tunable or command v1.38.0

09 **Read-only capability relay to dispatching host** NEW 35

ForgeDock adds a read-only capability relay from sandbox to dispatching host, with dispatch-time pre-fetch.

— Bare mechanism description with no further detail v1.38.0

10 **Need-based host capability pinning** IMPROVED 30

ForgeDock now pins host capabilities by actual need rather than by mention in the issue body.

— States a behavioural change with little mechanism detail v1.38.0

WAS THIS USEFUL? 👍 👎

planning-with-files

Feedback ⓘ v3.27.0 NOTES ↗ RANK

planning-with-files is a persistent file-based planning system for AI coding agents with crash-proof markdown plans, session recovery, and context-rot protection.

> planning-with-files adds a scriptable JSON inventory mode for planning files and aligns Claude plugin commands with the project's initializer and selection contracts.

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

01 **JSON inventory via `--list --json`** NEW 80

A new read-only JSON inventory mode exposes root and named planning-file paths, availability, phase metadata, shared defaults and explicit selection uncertainty. Available via `--list --json` on the CLI or `-List -Json` in PowerShell, e.g. `planning-with-files --list --json`.

Pull a machine-readable snapshot of all planning-file paths and phase metadata — useful for scripting or feeding a downstream agent.

```
$ planning-with-files --list --json
```

— Names exact flags and a runnable command with described output fields. v3.27.0

THINNER COVERAGE BELOW

02 **Claude plugin command contract alignment** IMPROVED 30

Claude plugin commands now follow the selected-plan and initializer contracts, initializing from the validated project root and delegating attestation to the shipped helpers.

— Describes behavior change but no concrete command or config surface. v3.27.0

WAS THIS USEFUL? 👍 👎

coffee-cli

Feedback ⓘ v3.8.0 NOTES ↗ RANK

coffee-cli is a clean AI terminal workspace with quick input, folder switching, copy & paste, and image pasting.

> Coffee CLI's v3.8.0 release adds automatic recovery for stuck desktop-mode replies, chat bubble styling for sent messages, file attachments in terminal tabs, and a clearer sidebar view toggle.

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

01 **Auto-recovery for stuck desktop mode replies** NEW 70

Coffee CLI now auto-ends a stuck process after 5 seconds in desktop mode and resumes the same conversation, preserving messages, model, thinking level, and permission mode — no force-quit needed.

— Clear mechanism and preserved state, but it's automatic with no user action. v3.8.0

THINNER COVERAGE BELOW

02 **File attachment via paperclip in terminal tabs** NEW 55

Regular terminal tabs can now attach file paths using the paperclip button (SSH remote tabs excluded).

— Names the exact UI control and the excluded scope. v3.8.0

03 **Chat bubbles for sent messages in desktop mode** NEW 40

Claude Code and Codex desktop mode now show sent messages as right-aligned chat bubbles, visually distinct from AI replies across all themes.

— Describes the visual change but it's a passive UI update. v3.8.0

04 **Sliding switch for history/workspace toggle** IMPROVED 25

The history and workspace toggle next to New Session in the sidebar now uses a sliding switch to show the current view more clearly.

— Minor UI tweak with limited detail beyond location. v3.8.0

WAS THIS USEFUL? 👍 👎

claude-mem

Feedback ⓘ v15.7.0 NOTES ↗ RANK

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

> claude-mem's latest release adds opt-in image-aware observations, configurable log retention, and broader credential redaction, alongside several security hardening changes to the worker and Postgres storage.

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

01 Image-aware observations in two-stage rollout NEW

88

Image-aware observations can now be enabled in two stages via the `CLAUDE_MEM_MEDIA_CAPTURE_ENABLED` and `CLAUDE_MEM_MEDIA_INFERENCE_ENABLED` environment variables (both default to false); the first phase adds bounded media contracts and a lazy Sharp 0.35.5 converter with size, pixel, and deadline limits.

Opt in to image capture and inference for a session where you want claude-mem to observe and summarise visual tool output.

```
$ CLAUDE_MEM_MEDIA_CAPTURE_ENABLED=true
CLAUDE_MEM_MEDIA_INFERENCE_ENABLED=true npx claude-mem install
```

— Names exact env vars, converter version and limits v15.7.0

02 Configurable worker log retention NEW

81

Daily worker log pruning is now controlled by `CLAUDE_MEM_LOG_RETENTION_DAYS` (default 30; set to 0 to keep forever); today's and yesterday's logs are always preserved regardless of the setting.

Trim worker logs after 7 days in a CI environment to avoid disk build-up, while keeping today's and yesterday's files unconditionally.

```
$ CLAUDE_MEM_LOG_RETENTION_DAYS=7 npx claude-mem install
```

— Exact env var, default value and runnable example v15.7.0

THINNER COVERAGE BELOW

03 Cross-origin write protection on worker endpoints IMPROVED

48

The worker now rejects cross-origin browser writes to session endpoints and to `POST /api/memory/ingest`.

— Names the endpoint but no reader action required v15.7.0

04 **Expanded token and credential redaction** IMPROVED 46

Redaction now covers GitLab, Hugging Face, and npm tokens, passwords embedded in URL credentials, and inbound sync text when local redaction is enabled.

— Lists redacted secret types but no config surface or example v15.7.0

05 **Postgres API key hashing and sslmode enforcement** IMPROVED 46

Postgres API keys are now stored as salted scrypt hashes, and every Postgres `sslmode` value is honored.

— Describes storage mechanism, no config example given v15.7.0

06 **Feedback button in My Work header** NEW 39

My Work now shows a Feedback button in its header, next to Star, that opens a My Work feedback issue form.

— UI location named but no deeper mechanism v15.7.0

WAS THIS USEFUL? 👍 👎

Tester Army e2e

Feedback ⓘ @e2e-dev/browserstack@0.1.0

NOTES ↗ RANK

Next generation e2e testing framework for web and mobile apps.

> e2e's biggest window addition is a new Agent Client Protocol package for testing coding agents, alongside a security-relevant breaking change that keeps screenshots active after secret fills, new mobile device-leasing and raw-session APIs, and a round of MCP tooling (drag_at, run_code, recording, param interpolation) plus CLI and locator improvements.

└──▷ **WHAT SHIPPED · 24 FEATURES** most completely described first ? what's the number?

01 **Multi-file download collection** NEW 85

browser.waitForDownloads(trigger, { count }) collects several files triggered from a single action, with completion and saving still bounded by timeout via the existing `waitForDownload`.

Collect multiple files triggered by a single button click, such as a bulk export, without writing per-file await chains.

typescript

```
const files = await browser.waitForDownloads(() =>
page.click('#export-all'), { count: 3 });
```

— Named API with count option and runnable example @e2e-dev/web@0.15.4

02 **Cloud device leasing for mobile** NEW 85

A new entry point `@e2e-dev/mobile/lease` exposes agentDeviceLeases() for device-provider packages to lease cloud devices, running one isolated daemon per worker slot and failing a release that agent-device cannot confirm ended.

Lease cloud devices from a provider package so each parallel worker gets its own isolated daemon session.

typescript

```
import { agentDeviceLeases } from '@e2e-dev/mobile/lease';

export default agentDeviceLeases({
  // provider-specific options
});
```

— Named entry point and function with runnable example @e2e-dev/mobile@0.12.5

03 **Raw agent-device session escape hatch** NEW 85

`device.unstable_raw` exposes the test's agent-device session name, the device's id and name, and a raw agent-device `client` bound to that session for operations the `device` methods do not cover; client calls are not recorded as steps, replayed from cache, or redacted.

Access raw agent-device internals for an operation the device API does not expose, such as calling a low-level client method directly in a test.

typescript

```
import { test } from '@e2e-dev/mobile';

test('raw session access', async ({ device }) => {
  const { sessionName, deviceId, deviceName, client } =
device.unstable_raw;
  console.log(`session=${sessionName} device=${deviceId}
(${deviceName})`);
  // call a client method not surfaced by the device API
  await client.someUnsupportedMethod();
});
```

— Named property fully described with usage example `@e2e-dev/mobile@0.12.5`

04 **Session-scoped screen recording** NEW 85

app.startRecording() and app.stopRecording() record a single screen span — a transition, a dialog opening — on web and on devices; `stopRecording` returns the video as a `path` or a `url` for hosted environments.

Record only the screen span that covers a dialog opening, without capturing the full test video.

typescript

```
test('the dialog opens', { video: 'off' }, async ({ app, screen
}) => {
  await app.open();
  await app.startRecording({ name: 'dialog' });
  await screen.getByRole('button', 'Open').tap();
  const [video] = await app.stopRecording();
  console.log(video.path ?? video.url);
});
```

— Named methods, return values, and a runnable example `e2e@0.22.0`

05 **Per-target browser permission grants** NEW 80

Setting `app.permissions` on a web() target grants browser permissions to every page of an attempt, for example `permissions: { 'clipboard-read': 'grant', 'clipboard-write': 'grant' }` for headless Chromium clipboard access.

Grant clipboard permissions to every page in a headless Chromium run so copy-paste flows work without prompts.

```typescript
export default web({
  app: {
    permissions: {
      'clipboard-read': 'grant',
      'clipboard-write': 'grant'
    }
  }
});
```

— Exact config key and values with code sample `@e2e-dev/web@0.15.4`

06 **Screenshots stay active after secret fills** BREAKING

80

`StepExecutorContext.pixelsTainted` is removed and a step's `visionDegraded` is never set to `PIXEL_TAINTED`, since filling a secret no longer disables screenshots for the model; values shown outside a secure field (e.g. behind a show-password toggle) now reach the model provider, so test credentials should be used for any secret filled into a plain input.

— Named fields, clear mechanism, migration guidance given `@e2e-dev/decision@0.3.1` `e2e@0.22.0`

07 **Route response inspection before fulfilling** NEW 75

route.fetch() lets a route handler read the upstream response before deciding how to fulfill or continue it.

Inspect and modify an upstream API response inside a route handler before the page sees it.

```typescript
await page.route('/api/data', async route => {
  const response = await route.fetch();
  const body = await response.json();
  await route.fulfill({ json: { ...body, injected: true } });
});
```

— Exact method named with working code sample `@e2e-dev/web@0.15.4`

08 **Agent Client Protocol package (@e2e-dev/acp)** NEW 70

`agent.act` and `agent.assert` run actions and assertions against your own coding agent over the Agent Client Protocol, signed in with the agent's own login; claudeCode() starts a Claude Code agent and codex() starts an OpenAI Codex agent, each with e2e's step tools as its only tools; any tool call the agent makes without asking fails the step with `POLICY_DENIED`, enforcing that all actions go through the runner so steps replay from cache without the agent.

— Full API surface named but no code example provided `@e2e-dev/acp@0.1.0`

09 **run_code tool for e2e mcp sessions** BREAKING 70

A `run_code` tool is added to `e2e mcp` sessions to execute test statements (locators, fills, assertions) on the live session's `screen` and `app`, returning each statement's result and what changed on screen; a project tool named `run_code` now fails config load with `INVALID_CONFIG`, so any existing project tool using that name must be renamed.

— Named tool and error code, migration step given, no example `e2e@0.22.0`

10 **params interpolation in agent.assert and agent.waitFor** NEW 70

`agent.assert` and `agent.waitFor` gain `params` support matching `agent.act`, so named values including unique() tokens can be interpolated into prompts with `{name}` syntax.

Pass a unique value through both an act step and an assert step to verify it appears on screen without repeating the literal string.

```typescript
const name = unique(`acme-${Date.now()}`);
await agent.act('create a company named {name}', { params: { name
} });
await agent.assert('the table shows company {name}', { params: {
name } });
```

— Named syntax and APIs with a runnable example `e2e@0.22.0`

11 **Negative locator filters** NEW 65

locator.filter() gains `hasNot`, which excludes matches whose subtree resolves a given locator, and `hasNotText`, which excludes matches whose text contains a string or matches a RegExp; third-party locator engines must handle both new fields.

Narrow a locator to elements that do not contain a specific text string, for example to skip 'Loading...' rows in a list assertion.

```typescript
const rows = page.locator('[data-testid="row"]').filter({
hasNotText: 'Loading...' });
await expect(rows).toHaveCount(5);
```

— Both named fields given, only one has an example `@e2e-dev/web@0.15.4` `@e2e-dev/mobile@0.12.5`

12 **drag_at pixel-coordinate drag tool** NEW 65

A `drag_at` tool is added to `agent.act` and `e2e mcp`, plus a dragAt(from, to) method on `ExecutorActions`, to drag between two screenshot coordinates for canvases, maps, or any element without a stable accessible id.

— Exact method signature named, no runnable sample `e2e@0.22.0`

13 **e2e init scaffolding options** IMPROVED 65

`e2e init` gains `--example with-angular` to scaffold a new project from the Angular standalone component example, and `--engine` now accepts a comma-separated list (e.g. `--engine web,mobile`), making the engine question a multiselect that generates a target and example file for each engine.

— Exact flags and example values, no code sample `e2e@0.22.0`

14 **Custom tool loop exports from e2e/agent** NEW 60

`e2e/agent` now exports `createGrammarTools`, `createVerdictTool`, `BASE_RULES`, `VERDICT_RULES`, `isRuntimeHardStop`, and the `ScreenOutput` type for building a custom tool loop that reuses the built-in actions and verdicts.

— Every export named, no usage example given `e2e@0.22.0`

THINNER COVERAGE BELOW

15 **fill/clear re-read and retry targets** NEW INTEGRATION IMPROVED 55

`fill` and `clear` now re-read targets before typing, including during cached replay, retrying after a fresh observation when inputs are missing, the security state has changed, or captures are incomplete; they also type into unclassified iOS elements that hold a value or only text, and Compose Multiplatform empty field placeholders folded into the iOS label are now read correctly as placeholders.

— Detailed mechanism but automatic, nothing to invoke `@e2e-dev/mobile@0.12.5`

16 **Separate pull request comments for e2e explore** IMPROVED 55

An `e2e explore` run now posts its own pull request comment instead of overwriting the comment from `e2e run`, so both results are visible side by side; a `--last-failed` rerun no longer folds in an exploration report that shares its output directory.

— Named commands and flag, behaviour clear, no example `@e2e-dev/github@0.4.4`

17 **open_session test parameter for e2e mcp** NEW 55

`e2e mcp`'s `open_session` gains a `test` parameter, named like an `e2e run` positional (e.g. `tests/checkout.e2e.ts:12`), to run that test before handing the session to the agent.

— Named parameter with example format, no code snippet `e2e@0.22.0`

18 **File-chooser triggers via button for uploads** IMPROVED 50

`upload` and `setInputFiles` now accept a button that opens a file chooser, including buttons placed over hidden file inputs.

— Named APIs but no runnable example `@e2e-dev/web@0.15.4`

19 **Device targeting skips held simulators/emulators** IMPROVED 50

A mobile target without a `device` specifier now skips booted simulators or emulators held by another agent-device session and drives free ones, reporting all holding sessions by name at `prepare` when none are free.

— Mechanism explained but no runnable step `@e2e-dev/mobile@0.12.5`

20 **e2e explore auto-runs init wizard** IMPROVED 50

When `e2e explore` runs in a project with no config, it now runs the `e2e init` wizard automatically instead of failing with `CONFIG_NOT_FOUND`.

— Named commands and error code, behaviour clear `e2e@0.22.0`

21 **open_app recovers the target app** IMPROVED 45

The `open_app` agent tool now returns to the target app when called without an app argument, letting the agent recover after a system sheet or another app takes the foreground.

— Named tool, behaviour described, no example `@e2e-dev/mobile@0.12.5`

22 **onBeforeAction hook for agent.act** NEW 45

`onBeforeAction` is added to `agent.act` as a hook that runs before each action so a test can capture the screen state an action started on.

— Named hook, purpose stated, no example `e2e@0.22.0`

23 **Enter-key press now taps unfocused field first** IMPROVED 40

press('Enter') on an unfocused field now taps it first before sending the key, matching the behaviour of character keys and fixing failures on iOS when the keyboard was down.

— Clear before/after but no example or config `@e2e-dev/mobile@0.12.5`

24 **Selective cookie clearing** NEW 35

Cookies can now be cleared selectively using validated name, domain, and path filters, leaving other browser storage intact.

— No method name or example given, mechanism vague `@e2e-dev/web@0.15.4`

└──▷ **BREAKING ON UPGRADE**

! `StepExecutorContext.pixelsTainted` is removed — any code that reads or writes this property will break.

! A step's `visionDegraded` is never `PIXEL_TAINTED` — code that branches on that value will no longer reach that branch.

! `StepExecutorContext.pixelsTainted` is removed; a step's `visionDegraded` is never `PIXEL_TAINTED`. Any code reading either field must be updated.

! Filling a secret no longer disables screenshots for the model — values shown outside a secure field (e.g. behind a show-password toggle) now reach the model provider. Use test credentials for any secret filled into a plain input.

! A project tool named `run_code` now fails config load with `INVALID_CONFIG`; rename any existing project tool that uses that name.

`@e2e-dev/browserstack@0.1.0` NOTES ↗ LATEST STABLE

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch @e2e-dev/browserstack@0.1.0
https://github.com/tester-army/e2e.git
# already have the repo? check out this version:
$ git checkout @e2e-dev/browserstack@0.1.0
```

WAS THIS USEFUL? 👍 👎

sirmalloc ccstatusline

Feedback ⓘ v2.2.33 NOTES ↗

RANK

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

> ccstatusline adds six new usage-tracking widgets covering spend limits, block forecasting, and Fast Mode, an opt-in developer hot-reload mode, and a live widget-picker preview, alongside a renamed/reworked Block Usage widget, clickable directory links, and a smarter `refreshInterval` default.

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

01 **Developer hot-reload mode** NEW 90

An opt-in developer hot-reload mode restarts the TUI in place during development: set `CCSTATUSLINE_DEV_RELOAD=1` and press Ctrl+R to reload the latest code while retaining unsaved settings and navigation state.

Reload ccstatusline with the latest code during development without losing unsaved settings or navigation state.

```
$ CCSTATUSLINE_DEV_RELOAD=1 npx ccstatusline
```

— Runnable command, env var, and shortcut all given v2.2.33

02 **refreshInterval default tuning** IMPROVED 85

`refreshInterval` now defaults to `10` seconds on fresh installs when Claude Code >= 2.1.97 is detected, and can be set anywhere from 1–60 seconds or removed entirely via the TUI.

— Named config key with exact default and range product docs

03 **New spend and block forecasting widgets** NEW 80

Adds six new widgets: Block Forecast and Block Limit Timer projecting the 5-hour usage block at reset and estimating time to the usage limit based on recent pace; Spend Limit Usage and Spend Limit Amount showing gateway-reported API spend percentages and dollar amounts without an extra API request; Fast Mode surfacing Claude Code's fast-mode state; and Extra Usage Limit displaying the monthly extra-usage limit — all with configurable formats and hide states.

— Names all six widgets and their behavior, but no config syntax v2.2.33

04 **Block Usage widget renamed and reworked** IMPROVED 70

The former `Session Usage` widget is renamed to `Block Usage`, labeled `5h:`, and now hides at 100%, while the `Block Reset Timer` widget hides below 100% and swaps the percentage display for a reset countdown once the 5-hour block is exhausted.

— Exact widget names, label, and hide thresholds given v2.2.33

05 **Status widget and glyph editor improvements** IMPROVED 65

The `claude-status` widget can optionally link to the Claude status page; the `remote-control-status` widget now supports editable Nerd Font on/off glyphs; and the TUI glyph editor adds a Shift+Tab shortcut to toggle any glyph slot between none and its default.

— Names three widgets/shortcuts but each described in one line v2.2.33

06 **Clickable directory links** IMPROVED 60

The `Current Working Dir` widget can now open the path directly — opening in a file manager, VS Code, or Cursor while keeping the shortened path label, with clickable links that open the path in the terminal.

— Named integrations but exact toggle/config unclear v2.2.33

07 **Per-profile usage data scoping** IMPROVED 60

Usage data and authentication are now scoped per `CLAUDE_CONFIG_DIR` when multiple Claude Code profiles are in use.

— Names the env var but no further mechanism v2.2.33

THINNER COVERAGE BELOW

08 **Fish-style path abbreviation for Git widgets** NEW 55

`Git Branch` and `Git Worktree` widgets now support fish-style path abbreviation via a toggle.

— Names the widgets and toggle but no exact key product docs

09 **Live widget picker preview** IMPROVED 50

The TUI's widget picker now shows a live preview of the highlighted widget while scrolling, before it's added, inserted, or changed, with scrolling picker lists.

— Describes behavior but no exact UI path or key v2.2.33

10 **Session Name falls back to generated title** IMPROVED

45

The Session Name widget now falls back to Claude Code's generated title when no custom rename exists.

— Clear behavior but minimal surface detail product docs

WAS THIS USEFUL?  

Amazon Kiro Crew

Feedback ⓘ v0.11.0 NOTES ↗ RANK

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

> Kiro Crew's v0.11.0 release overhauls mate architecture with per-mate agent files (Mate 2.0), makes KAS the default agent engine, and adds a batch of new CLI flags, config keys, and UI controls for scheduling, MCP persistence, and dashboard integrations.

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

01 **Per-mate agent files in Mate 2.0** NEW 75

Each Mate now owns an agent file under `~/.kiro/mates/agents/`, with a conductor chat and a worker session that follows its own Profile and skills; existing mates migrate automatically at startup.

— Mechanism and migration path described but no CLI/UI steps given v0.11.0

02 **CLI flags for schedule and agent control** NEW 75

Control schedules from the CLI with new flags: `kirocrew spawn run` now accepts `--model`, `--agent` and `--crew`; `kirocrew agent create` and `update` accept `--model`; `kirocrew cron add` and `update` accept `--auto-pause-after`.

Schedule a cron job to run at exactly the configured minute rather than drifting, and pause automatically after 30 minutes of consecutive failures.

```
$ kirocrew cron add --auto-pause-after 30
```

— Six named flags across three commands with a runnable example v0.11.0

03 **KAS set as default agent engine** IMPROVED 60

New installs default to the KAS engine, listed before Kiro CLI and shown as Kiro Agent; the desktop app ships the pinned KAS engine without requiring `kiro-cli`.

— Clear before/after but no exact command to switch v0.11.0

04 **Strict schedule start time for cron jobs** NEW 60

Set a cron job to start on the exact minute by adding `strict_schedule` to `app.json`.

Make a cron start on the minute by setting strict_schedule in app.json.

```json
{"cron": {"strict_schedule": true}}
```

— Exact config key and file shown with example v0.11.0

05 **Persistent MCP sign-in for remote servers** NEW 60

Keep remote MCP sign-ins across sessions by enabling `agent.kas_persist_mcp_sign_in` (off by default).

Persist remote MCP sign-ins across sessions so agents do not have to re-authenticate on restart.

```json
{"agent": {"kas_persist_mcp_sign_in": true}}
```

— Named config key and example provided, limited behavioural detail v0.11.0

THINNER COVERAGE BELOW

06 **Bitbucket Server PR support in dashboard** NEW 55

Read Bitbucket Server pull requests by listing hosts in `dashboard.bitbucket_server_hosts`.

— Named config key but no example or further mechanism v0.11.0

07 **Heartbeat check-in on Goals tab** NEW 55

Enable an opt-in Heartbeat check-in on the Goals tab to wake a mate every 30, 60, 120 or 240 minutes.

— Specific intervals and UI location given, no config key v0.11.0

08 **Run time and auto-pause limits for Schedule jobs** NEW 50

Set a run time limit and an auto-pause limit per Schedule job from the job form, with the board indicating when a job is in failure backoff.

— UI path named but no exact field names or values v0.11.0

09 **Per-mate tool requests in preview** NEW 50

Enable per-mate tool requests in preview by switching on the mate's own preview toggle (off by default), letting it find and request tools its worker lacks.

— Describes behaviour but no exact toggle name or location v0.11.0

10 **JSONL and NDJSON support in knowledge reader** NEW 35

Kiro Crew now supports `.jsonl` and `.ndjson` files in the knowledge file reader.

— Bare statement of supported formats, no further detail v0.11.0

WAS THIS USEFUL?  

SST OpenCode

Feedback ⓘ v2.2.3 NOTES ↗ RANK

OpenCode is an open-source AI coding agent for terminal-based development tasks.

> OpenCode's v2.2.x releases add Claude Code as a connectable agent with a guided setup flow, interactive HTML artifact publishing, persistent desktop terminals, and granular MCP/skill/plugin access-control rules, plus a percentage-based compaction buffer and several new worktree and composer shortcuts.

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

01 **Interactive HTML artifact publishing** NEW 90

Enable interactive artifact publishing by setting `"experimental": { "artifacts": true }` in `opencode.json`, letting agents publish, update, and serve HTML or static artifacts via a built-in local provider or plugin storage.

Allow an AI agent to publish and serve a live HTML report or static artifact during a session.

json

```
{
  "experimental": {
    "artifacts": true
  }
}
```

— Exact config key and file given with a runnable example. v2.2.4

02 **Percentage-based compaction buffer** NEW 90

Set a percentage-based compaction buffer with `"compaction": { "buffer": "20%" }` in `opencode.json` to reserve a share of each model's context window, covering models with both small and large windows.

Reserve 20% of any model's context window as a compaction buffer so a single config works across models regardless of their window size.

json

```
{
  "compaction": {
    "buffer": "20%"
  }
}
```

— Exact config key and runnable example provided. v2.2.5

03 **Claude Code agent connection** NEW 85

Connect to Claude Code models by installing Claude Code, running `claude auth login`, and selecting Claude Agent in `/connect` to reuse your existing Claude Code login. A guided setup flow now walks you through this: selecting Claude Agent from the provider picker copies `claude auth login`, prompts you to install Claude Code, and verifies the connection before proceeding to model selection.

Use Claude Code's authenticated session and models inside OpenCode without a separate API key.

```
$ claude auth login
# then inside OpenCode:
/connect
# select: Claude Agent
```

— Exact commands and UI flow make this immediately runnable. v2.2.4 v2.2.3

04 **OpenTUI 0.8 plugin compatibility requirement** BREAKING 80

`@opencode/plugin` now requires `@opentui/core` and `@opentui/solid` 0.8.0 or later. In OpenTUI 0.8, a `ScrollBox`'s `wrapper`, `viewport`, `content`, and `scrollbars` are facade objects instead of renderables, and a custom `RenderContext` must implement `clock` and `onAnimationTick`.

— Precise API changes named but requires plugin code migration to act. v2.2.3

05 **Granular access control for MCP, skills, and plugins** BREAKING 78

New `mcp.use`, `skill.use`, and `plugin.use` policy rule actions let you control which MCP servers, skills, and plugins are active. A bare `integration.use` rule no longer covers these resources, though existing rules with the old `mcp:`, `skill:`, and `plugin:` resource prefixes are converted automatically.

— Exact rule names and migration path given, but no runnable example. v2.2.3

06 **Base-branch chooser for worktrees** NEW 70

While creating a worktree, open the base-branch chooser with `⌘⌥.` on macOS or `Ctrl+Alt+.` on Windows and Linux, or reach it via the command palette and Settings › Shortcuts.

— Exact shortcut and menu path make it directly usable. v2.2.4

07 **Quote-send controls in comment popup** NEW 65

Send a single quote and comment from the comment popup by holding `⌘` or `Ctrl` to switch the checkmark to send, or pressing `⌘Enter` or `Ctrl+Enter`, without disturbing the rest of the composer.

— Exact key combinations named for a specific UI action. v2.2.5

08 **Shell mode toggle in composer** NEW 60

Toggle shell mode in the composer with `⌘⇧X` on macOS or `Ctrl+Shift+X` on Windows and Linux.

— Exact shortcut given but little explanation of behaviour. v2.2.5

THINNER COVERAGE BELOW

09 **Persistent desktop terminals** IMPROVED 50

Desktop terminals now belong to the session, survive server restarts with scrollback intact, and stay synchronized with the TUI and other windows; renamed terminals also keep their names across restarts.

— Describes behaviour change but no command or config to act on. v2.2.4

└──▷ **BREAKING ON UPGRADE**

! A bare `integration.use` rule no longer applies to MCP servers, skills, or plugins — you must now use `mcp.use`, `skill.use`, or `plugin.use` respectively (existing rules with the old `mcp:`, `skill:`, and `plugin:` resource prefixes are converted automatically).

! `@opencode/plugin` now requires `@opentui/core` and `@opentui/solid` 0.8.0 or later; in OpenTUI 0.8, a `ScrollBox`'s `wrapper`, `viewport`, `content`, and `scrollbars` are facade objects instead of renderables, and a custom `RenderContext` must implement `clock` and `onAnimationTick`.

WAS THIS USEFUL? 👍 👎

Amazon Kiro

Feedback ⓘ

Web: Session Recaps, Guided Workflows, and Reorganized Settings (2026-10-28)

BLOG ↗ RANK

Kiro is an agentic development environment that uses spec-driven workflows to plan, build, and maintain software.

> Kiro shipped session recaps and model fallback controls inside a reorganized Agent Config in Kiro Web, smarter guided workflows, and renamed Automations to Schedule, while Kiro CLI added an elevated-privilege startup warning and organization-enforced MCP/web-tool controls for V3 sessions.

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

01 Elevated-privilege startup warning in CLI NEW

80

Kiro CLI now warns on startup when running with elevated privileges (root on macOS/Linux, Administrator on Windows); disable this warning with `kiro-cli settings app.warnOnElevatedPrivileges false` when running elevated intentionally, such as in CI or containers.

Suppress the root/Administrator warning when you intentionally run Kiro CLI with elevated privileges in a CI pipeline or container.

```
$ kiro-cli settings app.warnOnElevatedPrivileges false
```

— Exact runnable command with clear use case. CLI: Elevated-Privilege Warning and Org…

02 New Agent Config sections: Session recap and Model fallback

NEW

67

`Agent Config > Memory` gains a Session recap section that summarizes the conversation after each message and shows a Recap area above the input after a few messages, and `Agent Config > Model fallback` lets you configure fallback behavior so a fallback model answers when Auto refuses a request, or continues from the visible response if the current model fails mid-reply.

Enable session recaps so each conversation is summarized automatically, making it easier to resume long agentic sessions.

📍 In the sidebar, go to Agent Config › Memory and turn on Session recap.

Configure a fallback model so work continues uninterrupted when Auto refuses a request or a model fails mid-reply.

📍 In the sidebar, go to Agent Config › Model fallback and select a fallback model from the list, which shows each model's context window such as `(1M context)`.

— Names exact config path and mechanism, with a usage example. Clear mechanism and exact navigation, but no edge-case detail.

Web: Session Recaps, Guided Workflows, …

03 **Sidebar reorganized into Agent Config and Settings** IMPROVED

65

The sidebar now separates `Agent Config` (Cloud Config, Steering, Skills, Workflows, Powers & MCP, Model fallback, Memory) from `Settings` (Account & plan, Appearance, Integrations, AWS access, Sandbox, API keys, Privacy & data, Help & support); MCP servers moved to `Powers & MCP`, the IAM role to `AWS access`, themes to `Appearance`, and model fallback to `Agent Config > Model fallback`.

— Names every relocated item but is navigation-only, no mechanism.

Web: Session Recaps, Guided Workflows, …

04 **Guided workflow check-ins and pass limits** IMPROVED

65

Workflows built from a prompt now open a `create-workflow` step and ask for missing details before planning, show each step's agent, model, effort, and first task, and pause after two rejected passes by default instead of running three to five passes and failing.

— Explains before/after behavior but no runnable surface.

Web: Session Recaps, Guided Workflows, …

05 **Organization MCP and web tool controls in V3 sessions** NEW

65

V3 sessions now turn MCP servers and the `web_search` and `web_fetch` tools on or off according to your organization's Kiro settings, and the `V3 /mcp` command lists configured servers that your organization's MCP registry hid along with instructions for declaring the ones the registry offers.

— Names specific tools and command but needs org admin context to use.

CLI: Elevated-Privilege Warning and Org…

THINNER COVERAGE BELOW

06 **Session reliability and state improvements** IMPROVED

55

Reopened sessions keep saved conversations readable and the composer usable while `Waking up sandbox` is shown, with messages sent during startup kept and a deleted sandbox now reporting its state immediately instead of after several minutes; each session now remembers whether its side panel is open, which tab it shows, its width,

and the selected workflow run and step; and at the daily session limit on GitHub, Kiro's reply now asks you to mention it again later.

— Describes concrete before/after behavior with no action needed from user. Clear list of remembered state but no interaction needed. Bare behavior note with no further detail.

Web: Session Recaps, Guided Workflows, …

07 **Automations renamed to Schedule** BREAKING 50

Kiro Web renames Automations to Schedule, with its tasks now called `scheduled tasks`; runs that hit concurrent-session or temporary-service limits now retry for a limited time instead of failing outright.

— Clear rename and retry behavior, but thin on retry limits.

Web: Session Recaps, Guided Workflows, …

08 **Context window shown in model pickers** IMPROVED 40

Model pickers and the fallback list now show each model's context window, such as (1M context).

— Small, self-contained UI detail with one concrete example.

Web: Session Recaps, Guided Workflows, …

09 **Skills file limit raised to 50** IMPROVED 35

Skills now support up to 50 files, up from 25.

— Exact numbers given but no mechanism or usage path.

Web: Session Recaps, Guided Workflows, …

10 **TUI fullscreen jump-to-prompt** IMPROVED 35

In the TUI fullscreen view, clicking the top row of a long answer now jumps back to the start of that prompt.

— Small UI behavior clearly described, nothing more to add.

CLI: Elevated-Privilege Warning and Org…

└──▷ **BREAKING ON UPGRADE**

! Automations is renamed to Schedule and its tasks are now called `scheduled tasks`.

WAS THIS USEFUL? 👍 👎

◆ Model Training & Fine-Tuning

fizgig

Feedback ⓘ v9.0.0 NOTES ↗ RANK

Fizgig — LoRA & Fine-tune Studio for Klein 9B, Krea 2, MiniMax H3, Qwen Image 2.1, Anima & Z-Image Turbo and Ace-Step XL: train, fine-tune, profile, repair and extract LoRAs & LoKR for Image, Video & Music Models

> Fizgig's latest releases add a DINOv2-based Style score for style LoRA previews, expand caption generation with dedicated style, concept and object/product task types, speed up training with progressive size warm-up, and automate RunPod pod login and checkpoint syncing alongside a raft of training-UI additions (live run summary, mid-run setting changes, a caption table view, sidebar monitoring).

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

01 **New Training Parameters options: progressive size warm-up and caption dropout**

NEW 90

Training Parameters gains two new options: a progressive size warm-up — enable Start at a smaller size to train 20%, 30%, or 50% of epochs at a reduced Target Megapixels (from 0.25 MP up) before blending to full size over a tenth of the run, which in a Z-Image Turbo test at 0.25 MP for the first 30% of a 1.0 MP run cut training time by about a fifth — and caption dropout via Other Options → Caption dropout (Off, 0.05, or 0.10), training that share of steps with no caption so the LoRA relies less on the trigger word.

— Exact toggle name, percentages and a measured speedup result Exact flag path and option values given

v8.9.98

02 **Captions tab task types and table view** NEW 85

On the Captions tab, caption tasks now cover every LoRA type — every caption length comes in a person/character version, an object/product version (covering shape, colours, materials, on-product text, placement, viewpoint and framing, in short and exhaustive-detail lengths), a style version, and a new concept version, 'Training caption (concept: a pose, action or effect)', with a The concept is field where you name the concept in plain words (e.g. a backflip or a double exposure) so every caption names it in those exact words after the trigger word, and Auto-recaption's First try uses and Second try uses settings now also offer the object variants alongside person variants — and captions can now also be viewed and edited as a table via the new View switch, showing thumbnail, file name, caption, word count, size, and mask, sortable by any column, while Find, Replace & Add now also adds words to the start or end of every caption.

— Names every caption type and field, with a clear UI entry point Names the switch and columns, UI path is clear but not a command

v9.0.0 v8.9.96 v8.9.98

03 **RunPod pod automation and checkpoint sync** IMPROVED 85

Fizgig now looks up a RunPod pod's password automatically using the RunPod API key saved on the Pods tab, so the app window and Helpers tab open pods already logged in without entering a password. The Move to... dialog offers a 'Bring its checkpoints and previews back here as they're saved' option (on by default) that syncs each checkpoint to the run's local folder as training proceeds on the pod, and a run moved to a pod now also carries the Samples tab's live preview settings so pod previews use the same configuration as the local machine. A pod helper whose login session expires now re-authenticates automatically without manual intervention.

— Names API key source, dialog option and tabs involved precisely v8.9.97

04 **Style score for style LoRA previews** NEW 78

A new Style score button in the Samples gallery scores style LoRA previews epoch by epoch, measuring texture, palette, line and medium against the 8 nearest training pictures and returning a 0–100% score per preview using DINOv2 (about 350 MB, CPU-only, downloaded on first use).

— Names mechanism, model and score scale but not exact navigation steps v9.0.0

05 **Live run summary panel on Training tab** NEW 65

A live run summary now appears under the run card on the Training tab, showing time, speed, epoch, peak VRAM, current learning rate, grad norm, loss, likeness score, and the full model/dataset/optimizer spec via Full spec .

— Lists every field shown but navigation is the only action v8.9.98

06 **Mid-run setting changes while paused** NEW 65

Active training settings can now be changed while a run is paused — learning rate (or Adaptive LR's Min and Max), save schedule, block swap, picture size, and Samples tab previews — all marked Applies on Resume .

— Names every adjustable setting and its resume behaviour v8.9.98

07 **Profiler seeds prompt from training run's Samples prompt** IMPROVED 60

Fizgig's Profiler now seeds its prompt from the first Samples prompt of the run that made the LoRA, so a style or object LoRA is profiled on the prompt actually used rather than a portrait; the bleed check uses the same prompt without the trigger word.

— Explains mechanism but behaviour is automatic, no user action named v8.9.96

THINNER COVERAGE BELOW

08 **Sidebar monitoring for disk, loss and background jobs** NEW

55

The sidebar now shows disk space, a per-run loss line, background job meters (captioning, model download, Image Prep, Profiler, extraction), and GPU load.

— Lists monitored items but no interaction beyond viewing v8.9.98

09 **Studio button on LoRA entries** NEW

55

A Studio button now appears on each LoRA entry on the Training tab, on Runs & outputs, and on each preview epoch that saved a LoRA, opening that LoRA in Studio on the model its file was trained for.

— Names the button and its three locations, a clear starting point v8.9.96

10 **Adaptive LR plateau detection on rolling average** IMPROVED

50

Adaptive LR now judges plateaus on the 3-epoch rolling average loss instead of a single epoch, and resumed runs carry the average forward.

— Describes the mechanism change but no UI path or config surface v9.0.0

11 **Per-type guidance across tabs** IMPROVED

45

Fizgig now surfaces plain per-type guidance across its tabs — face tools, Fast Identity Mode, Masked training, and the Subjects card — explaining what applies to a person, style, object or concept LoRA.

— Names the tabs covered but is purely explanatory text v8.9.96

12 **ACE-Step default preview prompt change** IMPROVED

25

ACE-Step's first preview prompt is now a song description instead of A high quality photo .

— Single-line default change with no further mechanism given v8.9.96

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

Arcade

Feedback ⓘ changelog · Oct 31, 2026 CHANGELOG ↗

Arcade provides an actions runtime that lets AI agents securely call external services with authorization and governance controls.

Arcade adds SMTP invitation email, per-server network allowlisting, secret withholding controls, worker resource sizing config, API key lifecycle management, and MCP Apps UI on by default.

Rotate, relabel, change expiration, or revive expired project API keys in place without recreating them.

Send invitation email through a self-hosted SMTP server by setting `coordinator.email` instead of relying on SendGrid.

Restrict deployed servers to a specific in-cluster destination without opening broader private-network access using the new network allowlist setting.

Withhold organization and platform secrets from tool servers registered or deployed through the API with `engine.workerSecretSharing`.

Set default resource requests and limits for deployed workers with `deployments.sizing.*`; use `deployments.resourceQuota.enabled` as the single quota on/off switch.

+8 more

RETIRED INTEGRATION Retired tools across Google Calendar, Workday, Zoom, and Slack

└──▷ BREAKING ON UPGRADE

- ! Google Calendar and Slack move to new major toolkit versions with new tool contracts and OAuth scopes — calls to tools by name will break; review Upgrade notes.
- ! Several toolkits rename tools to a consistent convention — calls to the old names will break; review Upgrade notes.
- ! Google Calendar's `DeleteEvent` and `FindTimeSlotsWhenEveryoneIsFree`, Workday's `ListTimeOffHistory`, Zoom's `ListUpcomingMeetings`, and several Slack tools with no alias are retired.
- ! Two required, additive engine migrations run automatically unless `engine.database.migrations` is `false`; rollback leaves added columns and indexes unused.
- ! `deployments.enabled: true` installs now require review of quota permissions and readiness assumptions before upgrading.

- ! The Helm values file is now schema-validated — review custom values before upgrading.
- ! Google and Microsoft sign-in now request additional OpenID Connect scopes; new connections will write a profile identifier.

WAS THIS USEFUL? 👍 👎

claw-orchestrator

Feedback ⓘ v7.16.1 NOTES ↗ RANK

Run Claude Code, Codex, Antigravity, Grok Build and OpenCode behind one runtime, and a run only counts as done when checks it didn't write pass. Cross-engine fan-out, councils, session handoff, spend caps, MCP server, ACP agent.

> claw-orchestrator's single v7.16.1 release adds per-run settings overrides for Antigravity sessions, registers a new Codex model, and fixes a session-naming collision in concurrent councils.

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

01 Per-run settings override for Antigravity sessions NEW

88

The `agy --settings` flag (requires agy 1.4.12) lets a run supply `settings.json` keys as inline JSON or a file path, applying a one-off override without touching global config; agy rejects any unrecognised key.

Apply a one-off settings override to a single Antigravity session without touching your global config.

```
$ agy --settings '{"timeout": 120, "retries": 3}'
```

Load per-run settings from a file when the inline JSON would be too long for the command line.

```
$ agy --settings ./overrides/session-settings.json
```

– Exact flag, requirement, input formats and runnable examples given. v7.16.1

THINNER COVERAGE BELOW

02 GPT-6.1 Terra registered for Codex NEW

45

Codex now supports GPT-6.1 Terra, priced at $1 / $5 per million input / output tokens and $0.05 per million cached tokens.

– Names model and pricing but no usage mechanism shown. v7.16.1

03 Fixed session collision in concurrent councils IMPROVED

45

Concurrent councils with members sharing the same name no longer collide: member session names now carry the complete council run id, keeping parallel runs' sessions

apart.

— Explains the bug and fix mechanism but offers no action for readers. v7.16.1

WAS THIS USEFUL? 👍 👎

pi-fabric

Feedback ⓘ v0.131.0 NOTES ↗ RANK

A programmable tool and agent runtime for Pi

> pi-fabric's v0.131.0 release adds new agent dispatch controls (model selection, queued spawning, named child skills, extension-provided model APIs), durable worker session persistence, and directional overflow indicators in the participant UI, alongside two breaking changes to actor name reuse and durable replay engine identity.

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

01 Directional overflow indicators in participant UI NEW

70

The UI now shows `↑ N more` on the Main row and `↓ N more` below the list where space permits, indicating hidden participants; compact layouts still preserve session identity, keyboard hints, and mouse targeting.

— Names exact indicator text, placement, and preserved UI behaviors. v0.131.0

02 Stopped actor name reuse now explicit BREAKING

70

A visible stopped actor now blocks creation of a same-name actor; to reuse the name, you must explicitly remove the stopped actor first, which deletes its retained session and history. This replaces the previous implicit removal that happened automatically during creation.

— States exact before/after behavior and the required migration step. v0.131.0

THINNER COVERAGE BELOW

03 Durable replay engine identity versioning BREAKING

55

The durable replay engine identity now includes the package version; journals written under a different engine identity are refused and will not be silently replayed.

— Describes the mechanism and refusal behavior but no migration path given. v0.131.0

04 Agent dispatch, model and skill selection NEW

40

Agents can now select models exactly, opt into queued agent spawning, choose named child skills, and use extension-provided model APIs; the release notes caution that exact model matching is not a final-dispatch security boundary for later hooks or extension-made calls.

— Four capabilities named but no flags, commands or config keys given. v0.131.0

05 **Durable worker session persistence** NEW 30

Adds durable worker session files, an embedded durable-host entry, and safer worker recovery and completion handling.

— Names the mechanism but gives no concrete config or command surface. v0.131.0

└──▷ **BREAKING ON UPGRADE**

! A visible stopped actor now blocks creation of a same-name actor; to reuse the name, explicitly remove the stopped actor first — removal deletes its retained session and history. This replaces the previous implicit removal during creation.

! The durable replay engine identity now includes the package version; journals written under a different engine identity are refused and will not be silently replayed.

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.8.223 NOTES ↗ RANK

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

> HyperFrames shipped frozen-frame detection in playback health reporting, re-landed reverse-video support, and added render-span exports plus standalone beat analysis for AI-agent video workflows.

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

01 Render a clip span as standalone MP4 NEW

85

The `render` command now accepts `--start` and `--end` flags to render a span of a film as its own standalone MP4, letting reviewers export a single scene without waiting for a full render.

Export only a clip window from a long film — useful for reviewing a single scene without waiting for a full render.

```
$ npx hyperframes render --start 00:01:10 --end 00:02:30
```

— Named flags with a runnable command example v0.8.222

02 Beat analysis on standalone audio/video files IMPROVED

75

The `beats` command can now run beat analysis directly on an audio or video file passed to it, without requiring a project context.

Detect beats in a standalone audio or video file to guide animation timing, without opening a project.

```
$ npx hyperframes beats ./music/intro-track.mp3
```

— Named command with a runnable usage example v0.8.222

THINNER COVERAGE BELOW

03 Render telemetry split into capture phases IMPROVED

55

The `render_complete` telemetry event now splits capture time into `seek`, `page`, `screenshot`, and `encoder wait` phases.

— Named event and phase fields but no access instructions v0.8.222

04 **Frozen-frame duration in playback health reports** NEW

35

Playback health reporting now reports how long a video on screen showed no moving picture, enabling frozen-frame duration tracking.

— Describes what it tracks but no API or threshold given v0.8.223

05 **Reverse-video re-landed on HyperFrames plan** IMPROVED

25

Reverse any video is now available on the HyperFrames plan, with clearer credit wording for the feature.

— Named but no mechanism or usage detail given v0.8.223

06 **Review notes name sub-composition review command** IMPROVED

25

Review notes now name the specific command that reviews cuts inside a sub-composition.

— Thin description with no named command or example v0.8.222

WAS THIS USEFUL? 👍 👎

PydanticAI

Feedback ⓘ v2.64.0 NOTES ↗ RANK

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

> PydanticAI v2.64.0 adds a ModelFallback capability with sticky routing and stream-timeout triggers, a BoxdSandbox workspace that runs the full harness toolset, and ArtifactScanner security scanning for agent memory and skills, alongside several breaking changes to error handling, server-state replay, and Coder defaults.

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

01 ModelFallback capability with sticky routing and timeouts

NEW 95

Adds a `ModelFallback` capability with a per-attempt `prepare_model_request` hook and `RetryModelRequest` for model-level fallback control, a `sticky` option to keep a conversation on the fallback model that answered, and a `first_event_timeout` model setting so a stream that never starts falls back to another model instead of hanging; `ModelFallback` can now also be constructed from agent specs.

Set a deadline for the first chunk of a model stream so a stalled provider triggers fallback automatically instead of blocking the run.

python

```python
from pydantic_ai import Agent
from pydantic_ai.models import ModelSettings
from pydantic_ai_harness import ModelFallback

agent = Agent(
    'openai:gpt-6.1-sol',
    model_settings=ModelSettings(first_event_timeout=10),
    capabilities=[
        ModelFallback('anthropic:claude-fable-5-1'),
    ],
)
```

Keep a conversation on whichever fallback model answered first, so subsequent turns stay consistent instead of re-trying the primary.

```python
from pydantic_ai_harness import ModelFallback

fallback = ModelFallback(
    'anthropic:claude-fable-5-1',
    sticky=True,
)
```

— Mechanism fully named with runnable fallback and sticky examples `v2.64.0`

02 **ArtifactScanner for Memory and Skills** NEW

75

Adds agent artifact security scanning for `Memory` and `Skills` with `ArtifactScanner`, which can be registered as a capability alongside other harness tools to scan agent-written artifacts before they are stored or used.

Scan agent-written artifacts in Memory and Skills for malicious content before they are stored or used.

```python
from pydantic_ai import Agent
from pydantic_ai_harness import Coder, Memory, ArtifactScanner

agent = Agent(
    'anthropic:claude-fable-5-1',
    capabilities=[
        Memory(),
        ArtifactScanner(),
        Coder(),
    ],
)
```

— Named capability with a runnable registration example `v2.64.0`

03 **Server state scoped to originating endpoint** BREAKING

70

Stored response, item, conversation, and container IDs are now sent back only to the endpoint that created them, controlled by `Model.accepts_server_state`; agents routing across multiple endpoints may stop replaying server state to mismatched endpoints.

— Mechanism and config key named with scope of impact `v2.64.0`

04 **ModelHTTPError raised on exhausted retries** BREAKING

65

A retrying transport that gives up now raises `ModelHTTPError` instead of the previous error, so fallback logic moves on; code catching the old exception type must be updated.

— Clear before/after behavior with migration note v2.64.0

05 **Coder strict tool schemas off by default** BREAKING 65

Strict tool schemas for `Coder` 's tools are now off by default; pass Coder(strict=True) to restore the previous behavior.

— Exact flag given to restore prior behavior v2.64.0

06 **Observability fields for Evals and online evaluation** NEW 60

Adds `logfire.scorer.score_type` and `logfire.scorer.run_id` fields to online evaluation result events, the OpenTelemetry `test.suite.name` and `test.case.name` attributes to Pydantic Evals spans, and CI provenance via OpenTelemetry `vcs.*` and `cicd.*` attributes on Pydantic Evals experiment spans.

— Several named telemetry fields but no usage guidance v2.64.0

07 **Dynamic system prompt append mode** IMPROVED 60

Lets @agent.system_prompt(dynamic=True) append changes with `on_change='append'` to preserve the cached prompt.

— Exact decorator and flag named, no runnable sample v2.64.0

08 **json-repair made an optional dependency** BREAKING 60

`json-repair` is now an optional dependency; installations that relied on it being present transitively must add it explicitly if they use `Coder` or `RepairToolArguments` .

— Exact migration action named for affected users v2.64.0

THINNER COVERAGE BELOW

09 **BoxdSandbox workspace** NEW 55

Adds `BoxdSandbox` : a workspace that runs `Coder` , `Shell` , `FileSystem` , and every other harness capability inside a boxd machine.

— Named capabilities included, no setup example v2.64.0

10 **output_speech_timestamps for realtime speech word timing** NEW 45

Adds `output_speech_timestamps` to report word timing of realtime speech on Azure AI Voice Live and ElevenLabs Agents.

– Named flag and providers, no example given v2.64.0

11 **Harness Reminder scope='conversation'** NEW 45

Adds Reminder(scope='conversation') for reminders that stay in the message history.

– Named option with minimal explanation of effect v2.64.0

12 **OnlineEvaluation runs once per durable run** BREAKING 45

`OnlineEvaluation` evaluators now run once per durable run instead of on every replay; evaluation counts will change for replayed runs.

– Behavior change stated, no migration example v2.64.0

13 **Planning plan shown in plan-tool results** BREAKING 45

The harness `Planning` plan is now shown in plan-tool results instead of a per-request reminder copy; any code or prompts that depended on the reminder copy will no longer see it.

– Behavior change noted, no concrete remediation given v2.64.0

14 **RunContext.load_capability for runtime capability loading** NEW 40

Adds `RunContext.load_capability` to load on-demand capabilities from code at runtime.

– States what it does, no mechanism or example v2.64.0

15 **ModelResponse.model_id field** NEW 35

Adds `ModelResponse.model_id` to record which model the agent sent each request to.

– Named field but no usage example shown v2.64.0

└──▷ BREAKING ON UPGRADE

! A retrying transport that gives up now raises `ModelHTTPError` instead of the previous error, so fallback logic moves on; code catching the old exception type must be updated.

! Stored response, item, conversation, and container IDs are now sent back only to the endpoint that created them, controlled by `Model.accepts_server_state`; agents routing across multiple endpoints may stop replaying server state to mismatched endpoints.

! `json-repair` is now an optional dependency; installations that relied on it being present transitively must add it explicitly if they use `Coder` or `RepairToolArguments`.

! Strict tool schemas for `Coder`'s tools are now off by default; pass Coder(strict=True) to restore the previous behavior.

! `OnlineEvaluation` evaluators now run once per durable run instead of on every replay; evaluation counts will change for replayed runs.

! The harness `Planning` plan is now shown in plan-tool results instead of a per-request reminder copy; any code or prompts that depended on the reminder copy will no longer see it.

WAS THIS USEFUL? 👍 👎

LangChain LangGraph

Feedback ⓘ 1.2.19 NOTES ↗ RANK

Low-level orchestration framework for building, managing, and deploying long-running, stateful AI agents.

> LangGraph 1.2.19 adds a small introspection improvement for ToolNode-based graphs.

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

01 Tool names exposed in graph node metadata IMPROVED

40

`ToolNode` tool names are now included in graph node metadata, allowing inspection of which tools a `ToolNode` exposes via graph introspection.

— Names the surface but no example or command shown 1.2.19

WAS THIS USEFUL? 👍 👎

aura

Feedback ⓘ v0.2.20 NOTES ↗ RANK

AURA is a production-tested SRE agent platform you can deploy in minutes. AURA handles the guardrails, APIs, state management, streaming, and failure handling required to put AI to work safely on production infrastructure.

> aura's v0.2.20 release adds a structured per-session event system—with lifecycle envelopes, a replayable event journal, resumed-run tracking and more forgiving deserialization—and now attaches token usage data to every terminal lifecycle event.

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

01 **Session event journal and lifecycle tracking** NEW 51

aura now gives each session an event envelope that carries run lifecycle events, enabling structured session-level event tracking. Each session also gets an event journal with replay, stored behind a capability in the session store. The event system lets a start event name the run it continues, supporting resumed-run tracking, and event deserialization now reads an unknown variant as `unknown` rather than failing, improving forward compatibility.

— Mechanism described but no API, flag or config key given v0.2.20

02 **Token usage on terminal lifecycle events** NEW 30

aura now attaches token usage data to every terminal lifecycle event, so consumption is visible at the end of each run.

— States the change but no metric schema or access point v0.2.20

WAS THIS USEFUL? 👍 👎

Microsoft Agent Framework

Feedback ⓘ python-1.24.0

NOTES ↗ RANK

The Microsoft Agent Framework builds, orchestrates, and deploys agents and multi-agent workflows in Python and .NET, consolidating AutoGen and Semantic Kernel.

> Microsoft Agent Framework's python-1.24.0 release adds experimental memory compaction and skill unloading, an NVIDIA NIM embedding client, async chat-client contexts across nearly every model provider, and async switch-case workflow conditions, alongside several breaking dependency and schema changes including a Redis context index migration and the end of Python 3.10 support.

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

01 Redis context provider index schema migration

BREAKING

85

`RedisContextProvider` (`agent-framework-redis`) now rejects the legacy raw-scope index schema; migration requires deleting the old context index and its context hashes, letting the provider create a new index, and re-ingesting context data — `overwrite_index=True` does not migrate incompatible indexes.

— Exact migration steps given, no single automated command python-1.24.0

02 Timeout parameter on FoundryChatClient

NEW

75

Adds a `timeout` parameter to `FoundryChatClient` in `agent-framework-foundry`, letting long-running model calls fail fast in latency-sensitive workflows.

Set a per-request timeout on a Foundry chat client so long-running model calls fail fast in latency-sensitive workflows.

python

```python
from agent_framework.foundry import FoundryChatClient
from azure.identity import AzureCliCredential

client = FoundryChatClient(
    credential=AzureCliCredential(),
    timeout=30.0,
)
```

— Runnable code example shows exact parameter usage python-1.24.0

03 In-run memory compaction and skill unloading

NEW

60

Adds experimental in-run memory compaction and skill unloading via the `AgentTriggeredCompactionStrategy` and `UnloadSkillsCompactionStrategy` classes in `agent-framework-core`.

— Classes named but no usage example provided `python-1.24.0`

04 **NVIDIA NIM embedding client** NEW 60

Adds an NVIDIA NIM embedding client in `agent-framework-nvidia`, along with the OpenAI-compatible embedding request-options extension point it requires in `agent-framework-openai`.

— Packages named, no code example given `python-1.24.0`

05 **Async chat-client contexts across providers** IMPROVED 60

Adds optional async chat-client contexts and ownership-aware connection cleanup across `agent-framework-core`, `agent-framework-anthropic`, `agent-framework-bedrock`, `agent-framework-foundry`, `agent-framework-foundry-local`, `agent-framework-gemini`, `agent-framework-mistral`, `agent-framework-ollama`, and `agent-framework-openai`.

— Lists every affected package but no API example `python-1.24.0`

06 **Python 3.10 support dropped** BREAKING 60

Python 3.10 is no longer a supported baseline; Python 3.11 or later is now required, and the next release will also remove Python 3.10 installation eligibility from Requires-Python.

— Clear version floor stated, no migration command needed `python-1.24.0`

07 **Minimum dependency version bumps across packages** BREAKING 60

Raises minimum dependency versions: `agent-framework-oracle` now requires python-oracledb 3.0 or later, `agent-framework-claude` now requires Claude Agent SDK 0.2.140 or later, and `agent-framework-azure-documentdb` and `agent-framework-mongodb` now require PyMongo 4.15 or later.

— Exact version floors named per affected package `python-1.24.0`

THINNER COVERAGE BELOW

08 **Default approval mode for local MCP tools** NEW 50

Adds an explicit default approval mode for local MCP tools and prompts that fall outside the per-tool approval lists, with a warning emitted when `never_require_approval` leaves the default implicit.

— Flag named but no full configuration example `python-1.24.0`

09 **Shared telemetry resource attribute** IMPROVED 50

Adds the shared `microsoft.agent-framework` telemetry resource attribute across logs, traces, and metrics, while preserving explicit resource overrides.

— Attribute named, no instrumentation example given `python-1.24.0`

10 **Agent Hooks middleware and FileAccessProvider prefixes** NEW 45

Adds the ability to load Agent Hooks middleware from a host declaration and adds support for tool-name prefixes on `FileAccessProvider`, both in `agent-framework-core`.

— Two thin additions named, no config example given `python-1.24.0`

11 **Async conditions in switch-case workflows** NEW 40

Supports async conditions in switch-case workflow edge groups in `agent-framework-core`.

— Brief description, no usage example shown `python-1.24.0`

└──▷ BREAKING ON UPGRADE

! The `RedisContextProvider` (`agent-framework-redis`) now rejects the legacy raw-scope index schema; delete the old context index and its context hashes, let the provider create a new index, and re-ingest context data — `overwrite_index=True` does not migrate incompatible indexes.

! Python 3.10 is no longer a supported baseline; Python 3.11 or later is now required. The next release will also remove Python 3.10 installation eligibility from Requires-Python.

! `agent-framework-oracle` now requires python-oracledb 3.0 or later.

! `agent-framework-claude` now requires Claude Agent SDK 0.2.140 or later.

! `agent-framework-azure-documentdb` and `agent-framework-mongodb` now require PyMongo 4.15 or later.

WAS THIS USEFUL? 👍 👎

AG-UI

Feedback ⓘ release/2026-10-30 NOTES ↗ RANK

AG-UI: the Agent-User Interaction Protocol. Bring Agents into Frontend Applications.

> AG-UI added a new WatsonX runs agent integration, subagent visibility controls for the Mastra adapter, and session lifecycle options for the ADK adapter, alongside several breaking changes to existing stream and session behavior.

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

01 WatsonX runs agent integration NEW

90

`WatsonxRunsAgent` in `@ag-ui/watsonx` connects to the `/v1/orchestrate/runs/stream` endpoint, surfacing run steps and raw tool responses as `TOOL_CALL_*` events. An optional `onToolResult` hook emits tool responses as `STATE_SNAPSHOT` events, backed by a pluggable thread store mapping AG-UI threads to watsonx threads.

Connect to the WatsonX orchestration runs endpoint and capture tool responses as state snapshots for inspection in the AG-UI frontend.

javascript

```
import { WatsonxRunsAgent } from '@ag-ui/watsonx';

const agent = new WatsonxRunsAgent({
  // connects to /v1/orchestrate/runs/stream
  onToolResult: (result) => {
    // emits STATE_SNAPSHOT events for each tool response
    console.log('tool result snapshot', result);
  },
});
```

— Named endpoint, event types, hook, and runnable code example given release/2026-10-30

02 Subagent visibility and tracking in Mastra adapter NEW

85

Setting `subagentVisibility: "hidden" | "attributed"` (default `"hidden"`) in `@ag-ui/mastra` controls whether sub-agent delegations appear as AG-UI subagent invocations; attributed mode maps each delegation to an AG-UI 1.0 subagent invocation. `RUN_FINISHED` now carries Mastra's finish reason on `metadata.mastra.finishReason` (cancelled runs carry none), and run usage now counts delegated sub-agents' token usage in both visibility modes.

— Exact config key, values, default, and metadata field named release/2026-10-30

03 Session lifecycle controls in ADK adapter

BREAKING

75

Passing `session_timeout_seconds=None` in `ag_ui_adk` now disables session expiry and prevents the cleanup task from starting, a change from prior behavior where the task still ran. Passing `track_sessions=False` (default `True`) skips the in-memory session index and cleanup task, but cannot be combined with `max_sessions_per_user`, which depends on that index.

— Two exact config flags with defaults and a stated incompatibility `release/2026-10-30`

04 WatsonX stream error handling and event ordering changes

60

BREAKING

In `@ag-ui/watsonx`, interrupted streams that previously ended as `RUN_FINISHED` now emit `RUN_ERROR`, requiring handling verification for partial runs. Tool response/argument queueing and flush timing also changed, which may alter the ordering and content of emitted `TOOL_CALL_ARGS` and `onToolResult` events.

— Names affected events but gives no migration steps `release/2026-10-30`

THINNER COVERAGE BELOW

05 AbstractAgent.middlewares visibility change in Mastra

40

BREAKING

`AbstractAgent.middlewares` in `@ag-ui/mastra` is now protected; subclasses that build their own clones must account for this access change.

— States the change but no migration mechanism given `release/2026-10-30`

└──▷ **BREAKING ON UPGRADE**

! `AbstractAgent.middlewares` in `@ag-ui/mastra` is now protected; subclasses that build their own clones must account for this access change.

! In `@ag-ui/watsonx`, interrupted streams that previously ended as `RUN_FINISHED` now emit `RUN_ERROR`; verify handling of partial runs.

! In `@ag-ui/watsonx`, tool response/argument queueing and flush timing changed, which may alter the ordering and content of emitted `TOOL_CALL_ARGS` and `onToolResult` events.

! In `ag_ui_adk`, `track_sessions=False` cannot be combined with `max_sessions_per_user`, which depends on the session index.

! In `ag_ui_adk`, `session_timeout_seconds=None` now disables session expiry and the cleanup task; previously the task still ran.

WAS THIS USEFUL?  

openexecutive

Feedback ⓘ v0.6.8 NOTES ↗ RANK

AI-powered virtual executive team — a single coherent executive persona backed by 8 specialist agents (FastAPI + Next.js).

> OpenExecutive v0.6.8 adds the ability to approve or decide pending cards directly from a chat app, improves file handling and search inside chat, and tightens briefing, mail, and delegated-draft housekeeping.

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

01 **File attach, split and save in chat** NEW 50

Any file can be attached, split, and saved by a single handle inside chat.

— Describes three distinct actions and the single-handle mechanism v0.6.8

02 **Briefing dismissal and replied-sender mail filtering** NEW 50

Users can dismiss a Right now briefing point, and mail from senders you have already replied to is dropped automatically.

— Names the specific briefing point but no exact UI path v0.6.8

03 **Automatic cleanup of delegated drafts** IMPROVED 40

Replaced chat drafts are automatically removed from a delegated person's Drafts folder.

— Explains the automatic mechanism but offers no reader action v0.6.8

04 **Approval card decisions from chat app integrations** NEW 35

Users can decide every waiting approval card directly from a chat app via the integrations feature, rather than returning to the main interface.

— Names the integrations feature but no specific chat apps or commands v0.6.8

05 **Reuse of prior chat search before new lookups** IMPROVED 33

The system now searches a speaker's own earlier chats before making new lookups, reducing redundant queries.

— Explains behavior but is an internal change with no user action v0.6.8

WAS THIS USEFUL? 👍 👎

AWS Strands Agents

Feedback ⓘ harness-typescript/v0.3.3

NOTES ↗ RANK

Strands Agents is AWS's open-source SDK for production agents in Python and TypeScript, with a model-driven loop, tool integration, and deployment targets.

> Strands Agents brought its graph, vended file, and DecisionModel tooling to TypeScript (porting from Python), added an S3 Vectors search strategy for S3Storage memory, and shipped breaking changes to TypeScript multi-agent interrupt handling and tool streaming, alongside new Python APIs for runtime tool removal, Graph/Swarm turn-and-token limits, and model introspection.

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

01 **New TypeScript tools ported from Python** NEW 70

TypeScript gains several tools ported from Python: `read` and `write` tools for reading and writing files, an `edit` tool for in-place file editing (deprecating the previous `fileEditor`), and a `graph` tool for multi-step traversal workflows.

Use the `read`, `write`, and `edit` vended tools together so an agent can read a config file, patch it, and write the result back.

python

```python
from strands import Agent
from strands.vended_tools import read, write, edit

agent = Agent(tools=[read, write, edit])
agent('Read /etc/app/config.yaml, change the log level to DEBUG,
and write the file back')
```

— Names the exact tools and the deprecated predecessor, with a usage example Describes purpose but no mechanism, flag, or example

typescript/v1.23.0

THINNER COVERAGE BELOW

02 **DecisionModel primitive for decision-making** NEW 55

TypeScript can now accept a `DecisionModel` as a `HumanInTheLoop` risk classifier to route decisions programmatically, and Python gains a `DecisionModel` primitive plus `LLMDecisionModel` for composable decision-making logic.

— Names the primitives but no usage example or code typescript/v1.23.0

03 **TypeScript interrupt handling now persists on parent agent**

55

BREAKING

Subagent and swarm interrupt handling in TypeScript has changed so that interrupts are now persisted on the parent agent, which may alter existing interrupt-handling logic in TypeScript multi-agent workflows; related to this, an agent invocation can be cancelled mid-run.

Cancel a long-running TypeScript agent mid-invocation, for example when a user clicks a stop button in a UI.

typescript

```
const agent = await createHarness()
const run = agent.invoke('Analyze every file in the monorepo')
// Later, e.g. on user request:
agent.cancel()
```

— Describes the behaviour change and includes a cancellation example `typescript/v1.23.0`

04 **Automatic Bedrock maxTokens defaults** IMPROVED 50

Bedrock model calls now default `maxTokens` to the value reported by the model's output limit, removing the need to set it manually for each model.

— Names the field and behaviour change but no example `harness-typescript/v0.3.3`

└──▷ **BREAKING ON UPGRADE**

! Subagent and swarm interrupt handling in TypeScript has changed: interrupts are now persisted on the parent agent, which may alter existing interrupt-handling logic in TypeScript multi-agent workflows.

! Shared agent tool streaming logic has been consolidated, which may break existing code relying on prior streaming behaviour in TypeScript agent tools.

WAS THIS USEFUL? 👍 👎

cotal

Feedback ⓘ v0.131.2 NOTES ↗ RANK

The open standard for agent coordination

> cotal's latest releases focus on trust and clarity in agent coordination: `cotal pending` now shows broker-verified sender identity instead of self-reported IDs, while CLI guidance, terminal output, and backup/restore tooling all got smaller usability and API-contract fixes.

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

01 Backup cleanup-fallback discard-history flag NEW

82

The `cotal backup cleanup-fallback --discard-history` command lets you safely preview what a cleanup would remove after a restore, and only proceeds once you accept losing history in the previous store.

Safely preview what a cleanup would remove after a restore, and only proceed if you accept losing history in the previous store.

```
$ cotal backup cleanup-fallback --discard-history
```

— Exact runnable command with flag and behaviour v0.128.0

02 Broker-verified sender principal in `cotal pending` BREAKING

75

`cotal pending` candidate output now displays the broker-bound sender principal taken from the message subject in the `from` field, replacing the self-reported sender ID previously written inside the message itself, in both JSON and plain output formats. Candidates whose subject cannot be parsed are withheld as unreadable rather than displayed.

— Names exact field and formats but no example shown v0.131.2

03 Clearer no-manager guidance across CLI commands IMPROVED

70

When `cotal ps` refuses to run because no manager is active, it now names the root of the machine's own stack for the space and suggests running bare `cotal up` in that folder; `cotal status <agent>` and the `attach` picker share this same improved guidance.

— Names affected commands but no exact message text v0.128.0

04 Restore-summary helpers require mode field

BREAKING

66

Exported restore-summary helper functions now require the record's `mode`; library callers that previously passed a partial record must now pass the validated maintenance journal including its `mode`. This is a breaking TypeScript API change.

— Names required field and API impact, no migration example v0.128.0

THINNER COVERAGE BELOW

05 **Wider terminal wrapping for `cotal up` startup notices**

IMPROVED

56

`cotal up` now wraps four more startup notices to fit the terminal width — space selection, kept selection, `wrote default persona`, and new-project space naming — leaving paths wider than the line unsplit.

— Names the four notices but is a cosmetic CLI tweak v0.128.0

06 **Sender and message IDs in Codex injected messages** NEW

30

cotal adds sender and message IDs to Codex injected messages and inbox replies to distinguish between senders that share the same name.

— Only described in summary line, no mechanism or example v0.128.0

└──▷ BREAKING ON UPGRADE

! The `from` field in `cotal pending` output now shows the sender principal from the broker-bound message subject instead of the sender ID written inside the message; candidates whose subject cannot be parsed are withheld as unreadable rather than displayed.

! The exported restore-summary helpers now require the record's `mode`; library callers that previously passed a partial record must now pass the validated maintenance journal including its `mode`. This is a breaking TypeScript API change.

WAS THIS USEFUL? 👍 👎

busabase

Feedback ⓘ v1.0.0 NOTES ↗ RANK

Open-source database & workspace for AI agents — structured data, durable knowledge, reusable skills, runnable apps, and human review on the writes that matter. Local-first and self-hostable.

> busabase shipped its v1.0.0 release, publishing the full toolchain — server, CLI, and SDK packages — to npm and Docker Hub.

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

01 **v1.0.0 server, CLI, and SDK packages** NEW 85

Run the busabase server with `npx busabase@1.0.0 server` or the multi-arch (amd64/arm64) Docker image `busabase/busabase:1.0.0`, mounting data at `/data`. Interact with a running server via the API client CLI `npx busabase-cli@1.0.0` (alias `busa-cli`), or consume it from Node.js using the published `busabase-sdk` and `busabase-cms-sdk` npm packages (alias `busa-sdk`).

Start the busabase server locally with persistent storage for immediate use.

```
$ docker run --rm -p 15419:15419 -v ~/.busabase/data:/data
busabase/busabase:1.0.0
```

Explore all available CLI operations against a running busabase server.

```
$ npx busabase-cli@1.0.0 --help
```

— Exact commands, package names, and Docker image given; no deeper behaviour described

v1.0.0

WAS THIS USEFUL? 👍 👎

Vercel AI SDK

Feedback ⓘ ai@7.0.158 NOTES ↗ RANK

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

> Vercel AI SDK extended Experimental_SandboxSession with new file-system inspection and cleanup methods.

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

01 File-system methods on Experimental_SandboxSession NEW

81

Sandbox sessions gain `stat`, `listDirectory`, and `deleteFile` methods for file-system inspection and deletion, with relative path resolution, implemented in both the just-bash and Vercel Sandbox adapters.

Inspect a file, list a directory, and clean up a generated artifact inside a sandbox session after an agent task completes.

typescript

```typescript
// After running a task in the sandbox session
const info = await session.stat('output/report.json');
const files = await session.listDirectory('output');
await session.deleteFile('output/tmp-scratch.json');
```

— Names exact methods and adapters with a runnable usage example. ai@7.0.158

WAS THIS USEFUL? 👍 👎

yoagent

Feedback ⓘ v0.26.2 NOTES ↗ RANK

yoagent is a Rust agent loop that streams from 7 LLM protocols and runs tools until done, from a laptop to a Cloudflare Worker.

> yoagent's latest release focuses entirely on making context compaction safer, ensuring the task prompt and in-flight tool calls survive tight budgets.

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

01 **Safer context compaction ordering** IMPROVED 71

Context compaction now preserves, in order, the head, the latest user message, the newest turn, and as much of the remaining tail as fits, overshooting the target budget rather than dropping the task prompt first. When compaction cuts past the latest user message on the normal path, that message is retained and placed right after the `[Context compacted]` marker. The kept head is now guaranteed to never end partway through a multi-call turn's results, preventing unanswered tool calls after compaction, and the guaranteed tail is never less than the newest turn, so a small `keep_recent` value no longer drops the turn whose results the model was waiting on.

— Explains mechanism and names keep_recent and marker, but no code example given. v0.26.2

WAS THIS USEFUL? 👍 👎

lexmount Moli

Feedback ⓘ v1.1.20 NOTES ↗ RANK

Best headless browser for AI agents. Lite, Fast, High-Compatibility. Built in Rust

> Moli reworked IndexedDB to use a SQLite-backed storage and transaction model, added curl-style cookie jar support to Fetch, extended CDP with version exposure and drag interception, and added a WebIDL Clamp conversion.

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

01 **CDP version exposure and drag interception** NEW 80

CDP now exposes `Moli-Version` in `/json/version`, and adds `Input.setInterceptDrags` and `Input.dragIntercepted` for drag event interception.

Check the running Moli version programmatically via CDP — useful in automation scripts that need to gate behaviour on the browser version.

```
$ curl http://localhost:9222/json/version
```

— Exact endpoint, field and CDP methods with a runnable curl example. v1.1.20

02 **IndexedDB storage and transaction overhaul** IMPROVED 68

IndexedDB now stores data in SQLite instead of JSON, with transactional schema renames supported and transactions admitted by scope and mode. It also gains native geometry snapshots with shared codecs, `DOMRect` snapshots with shared codecs, FIFO connection request leases, versionchange acknowledgements, and support for native records and bulk query options.

— Mechanism and scope named, but no API or command to invoke it. v1.1.20

THINNER COVERAGE BELOW

03 **Curl-style cookie jar support in Fetch** NEW 40

Fetch now supports curl-style cookie inputs and jar export.

— Named but no concrete syntax or options shown. v1.1.20

04 **WebIDL Clamp unsigned long conversion** NEW 25

WebIDL adds Clamp unsigned long conversion.

— Bare name with no further detail given. v1.1.20

WAS THIS USEFUL? 👍 👎

truefoundry TrueForge

Feedback ⓘ

`@truefoundry/trueforge-ui@0.182.0` NOTES ↗ RANK

The open-source agent harness - the runtime layer that turns an LLM into a working agent.

> TrueForge ships a session sharing and grants API for per-user and tenant-wide access, adds inline per-request MCP tool definitions via `x-tfg-tools`, and improves chat visibility with context-compaction and live agent-status indicators, alongside a breaking change removing the `last_turn_id` session column.

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

01 **Inline MCP tool definitions per request** NEW 85

Callers can supply inline MCP tool definitions per request via the `x-tfg-tools` header on turn requests, injecting tool schemas dynamically at turn time without pre-registering them.

Supply inline MCP tool definitions on a per-request basis without pre-registering them, so a caller can inject tool schemas dynamically at turn time.

```
$ curl -X POST https://<host>/turns \
  -H 'Content-Type: application/json' \
  -H 'x-tfg-tools: [{"name": "search", "inputSchema": {"type":
"object", "properties": {"query": {"type": "string"}}},
"outputSchema": {"type": "string"}}]' \
  -d '{"session_id": "<session-id>", "message": "Search for
recent CVEs"}'
```

— Runnable curl example with exact header name `@truefoundry/trueforge-core@0.182.0`

02 **Session sharing and access control API** NEW 80

Sessions can now be shared with specific users via invite search, grant, and revoke actions when the `session_sharing.users` capability is reported, backed by a new session grants API for sharing, listing, and revoking access. Tenant-wide sharing continues to use the `session.shared` flag and is surfaced via the `session_sharing.enabled` capability; tenant session grants are omitted from the session access collaborator list so it only shows per-user grants.

Grant a specific user access to a session using the new per-user sharing capability when `session_sharing.users` is reported.

📍 In the session detail view, open the Share panel, search for the user by name or email in the invite search field, select them, and click Grant. To revoke, find the user in the collaborator list and click Revoke. Tenant-wide access is still toggled via the `session.shared` flag separately.

– UI steps and capability names given, but no API endpoint path named

`@truefoundry/trueforge-ui@0.182.0` `@truefoundry/trueforge-core@0.182.0`

---

03 **Scheduled session visibility in session list** NEW 65

UI-created sessions are stamped with `tfg.source: ui` metadata, and Chat History is filtered on that field so scheduled sessions are hidden by default; the session list tooltip also links each scheduled session to its schedule by id.

– Names the metadata field but no endpoint to manage it `@truefoundry/trueforge-ui@0.182.0`

---

04 **Session tip now read from last_turn_details.id** BREAKING 60

The `last_turn_id` column is removed from sessions; the session tip is now read from `last_turn_details.id` instead.

– Names exact column change but no migration steps `@truefoundry/trueforge-core@0.182.0`

THINNER COVERAGE BELOW

05 **Size limits on large tool responses** NEW 55

Large tool responses now include the content size in the blocked preamble and state per-response character limits for dumped-file reads and oversized sandbox stdout, phrased using RFC 2119 (`MUST` / `MAY` / `SHOULD`) wording.

– Names wording convention but not exact limit numbers

`@truefoundry/trueforge-core@0.182.0`

---

06 **Context compaction visibility in chat** NEW 50

Chat agent steps now display context compaction as `Summarizing` and `Summarized context` with expandable event JSON, and the session event timeline marks a Summary entry for it.

– Describes UI labels but no config or API surface `@truefoundry/trueforge-ui@0.182.0`

07 **Live agent status in chat loader** IMPROVED 50

The chat loader now shows what the agent or sub-agent is currently doing instead of a static "Working..." message, including during context summarization, displays Taking longer than expected after a minute without new stream content, and shows a step count on sub-agent cards.

— Clear behavior change but no named config or flag `@truefoundry/trueforge-ui@0.182.0`

08 **Model catalog warning in model pickers** NEW 35

The composer and agent-config model pickers now surface a model catalog warning when reasoning effort or context length is not configured.

— One-line description with no further mechanism `@truefoundry/trueforge-ui@0.182.0`

09 **In-progress tip turn rendering** NEW 30

Session detail now renders in-progress tip turns from already-loaded events and indicates that the turn may still be incomplete.

— Brief description with no mechanism or config named `@truefoundry/trueforge-ui@0.182.0`

NEW INTEGRATION EvoLink remote MCP server preset

└──▷ BREAKING ON UPGRADE

! The `last_turn_id` column is removed from sessions; the session tip is now read from `last_turn_details.id`.

`@truefoundry/trueforge@0.182.0` NOTES ↗ STABLE

TrueForge adds a session grants API (share, list, revoke), Sentry error reporting via `VITE_SENTRY_DSN`, and per-request MCP tool injection via `x-tfg-tools`.

└──▷ GET THIS VERSION ▾

pipx

```
$ pipx install trueforge-sdk==0.182.0
# already installed? upgrade to this version:
$ pipx install --force trueforge-sdk==0.182.0
```

npm

```
$ npm install -g @truefoundry/trueforge@0.182.0
```

```git
$ git clone --branch @truefoundry/trueforge@0.182.0
https://github.com/truefoundry/trueforge.git
# already have the repo? check out this version:
$ git checkout @truefoundry/trueforge@0.182.0
```

PyPI attestation · signed build provenance · checked 31 Oct Wrong command?

Enable optional Sentry error reporting on the frontend by setting the `VITE_SENTRY_DSN` build arg; traces session-create and turn-stream API calls.

Send a session ID to the gateway on model and MCP calls using the `x-tfy-conversation-id` header alongside `tfg.session_id` metadata.

Inject per-request MCP tool definitions — including `inputSchema` and optional `outputSchema` — by passing `x-tfg-tools` as an array on a TrueFoundry-mode turn.

TrueForge now adds a session grants API supporting share, list, and revoke operations, plus user listing.

Session grants now give read access; sharing with everyone in the tenant uses a `tenant` grant, surfaced as `Session.shared`, and the `session_sharing.enabled` capability flag is added. The deprecated `PATCH /sessions/{id} { shared }` now grants or revokes the tenant grant.

└──▷ **BREAKING ON UPGRADE**

! The `last_turn_id` column is removed from sessions; the session tip is now read from `last_turn_details.id` — any query or code referencing `last_turn_id` directly will break.

WAS THIS USEFUL? 👍 👎

trigger.dev

Feedback ⓘ v4.8.3 NOTES ↗ RANK

Trigger.dev – build and deploy durable AI agents and workflows

> trigger.dev v4.8.3 adds CLI tooling for rotating webhook signing secrets, an early-refresh API for Convex-tracked runs, private-beta OIDC authentication for Cloud tasks, and machine-bound remote CLI login, while deprecating the legacy Bun runtime ahead of a 2027 rejection date.

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

01 Webhook secret rotation CLI commands NEW 90

`trigger webhooks set-secret` reads a new signing secret from a pipe, environment variable, or `.env` file without exposing it in shell history, and `trigger webhooks generate-secret` creates a new secret and saves it to an env file. Both commands accept `--previous-secret-ttl <duration>` to control how long the old secret keeps working during rotation.

Rotate a webhook signing secret in CI without the new value ever appearing in shell history or logs.

```
$ echo "$WEBHOOK_SECRET" | trigger webhooks set-secret --previous-secret-ttl 24h
```

— Exact command names, flag, and a runnable rotation example given v4.8.3

02 Deprecation of legacy Bun runtime DEPRECATED 90

`runtime: "bun"` and its alias `runtime: "bun-legacy"` (Bun 1.3.3 with Node.js 20) are deprecated: deploys using either value now print a warning and will be rejected starting January 5, 2027. Projects should switch to `runtime: "bun-1.4"` (Bun 1.4.4, no Node.js binary) in `trigger.config.ts`.

Switch a deploy off the deprecated Bun legacy runtime before the January 2027 rejection deadline.

```yaml
runtime: "bun-1.4"
```

— Exact config values, versions, and a hard rejection date given v4.8.3

03 Run refresh API for Convex integration NEW 75

Calling trigger.refresh(ctx, id) in `@trigger.dev/convex` requests an earlier status check for a Convex-tracked run, letting Trigger Realtime applications publish results without waiting for the next scheduled poll.

Push a Convex-tracked run's result to subscribers immediately rather than waiting for the next poll interval.

typescript

```
import { trigger } from "@trigger.dev/convex";

// Inside a Convex action or mutation
await trigger.refresh(ctx, runId);
```

— Named function and package with a short runnable code example v4.8.3

---

04 **Remote CLI login with code, machine-bound sessions** BREAKING

70

`trigger.dev login --code <code>` lets a user log in from a remote machine when the browser and terminal are on different machines. CLI and MCP logins are now bound to the machine that started them, which means older CLI versions can no longer complete `trigger.dev login` and must upgrade first.

— Exact flag given plus the breaking impact on older CLIs v4.8.3

---

05 **OIDC authentication for deployed Cloud tasks** NEW 65

A private-beta feature enables OIDC authentication for deployed Cloud tasks via `@trigger.dev/sdk/oidc/aws`, backed by an AWS credential provider. Using it requires installing `@aws-sdk/client-sts` and redeploying after upgrading the CLI.

— Named package and setup step but no usage example, still beta v4.8.3

└──▷ **BREAKING ON UPGRADE**

! Older CLI versions can no longer log in after the login-binding change — upgrade to v4.8.3 before attempting `trigger.dev login`.

! `runtime: "bun"` and `runtime: "bun-legacy"` deploys will be rejected from January 5, 2027; migrate to `runtime: "bun-1.4"` and redeploy before that date.

WAS THIS USEFUL? 👍 👎

Feynman

Feedback ⓘ v0.5.61 NOTES ↗ RANK

Feynman is a personal AI research assistant that searches papers and the web, builds literature reviews, and produces cited research reports.

> Feynman now lets users manage MCP servers directly from the CLI, with definitions persisted to disk and loaded automatically into every session.

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

01 MCP server management via feynman mcp commands

BREAKING

93

New `feynman mcp add`, `feynman mcp list`, and `feynman mcp remove` commands run Pi's own MCP machinery under Feynman's settings, saving server definitions to `~/.feynman/agent/mcp.json` and loading them automatically at the start of every session. This changes `feynman mcp list` behaviour: it no longer forwards the query to the model as a chat question but instead runs Pi's MCP list command directly, so any workflow relying on the old chat-passthrough behaviour will break.

Wire a local Zotero library into every Feynman research session via the zotero-mcp server.

```
$ feynman mcp add zotero --env ZOTERO_LOCAL=true -- zotero-mcp
```

Confirm which MCP servers are registered and will load at the start of the next session.

```
$ feynman mcp list
```

— Names exact commands, config file path, and documents the breaking behaviour change.

v0.5.61

└──▷ **BREAKING ON UPGRADE**

! `feynman mcp list` no longer forwards the query to the model as a chat question — it now runs Pi's MCP list command directly. Any workflow that relied on the old chat-passthrough behaviour will stop working.

WAS THIS USEFUL? 👍 👎

openclacky

Feedback ⓘ v1.5.22 NOTES ↗ RANK

The most Token-efficient open-source AI Agent

> openclacky v1.5.22 adds Kimi K3.1 Pro model support, a rebuilt Git changes panel with scoped review and lazy diffs, and a batch of composer and preview UI improvements.

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

01 Git changes panel rebuild with scoped review IMPROVED

80

Rebuilds the Git changes panel with scoped review and lazy diffs, including a committed scope pill for choosing which unpushed commits to review and which base to compare against. Adds a per-file menu from the diff header, independent sideways scrolling for each diff pane with gutters pinned, and a single-row toolbar to toggle the file tree and expand all nodes.

– Detailed mechanism and named UI elements, no exact command v1.5.22

02 Composer drag-and-drop, history recall, and panel toggle NEW

60

Drag workspace files straight into the composer to attach them to a message. Recall previously sent messages in the composer using the arrow keys. Toggle the aside panel with `Cmd/Ctrl+L`, with the shortcut shown in the panel's tooltip.

– Lists three distinct interactions, one with exact shortcut v1.5.22

THINNER COVERAGE BELOW

03 Image preview zoom, pan, and attachment viewing NEW

55

Adds zoom and pan for image previews, the ability to open attachment cards directly in the file preview, and a chrome-style preview toolbar with zoom controls.

– Describes new preview controls, no exact UI path v1.5.22

04 Kimi K3.1 Pro model support NEW

50

Adds Kimi K3.1 Pro and K3.1 Pro Highspeed models, and sets the Kimi provider's default model to K3.1 Pro.

– Names specific models but no usage detail v1.5.22

05 Branch picker on new session page NEW

45

Adds a git branch picker to the new session page so you can choose the starting branch before a session opens.

— UI starting point named, no mechanism detail v1.5.22

06 **Redesigned scheduled tasks page** IMPROVED 35

Redesigns the scheduled tasks page and shows when each scheduled task runs next.

— Thin description, no mechanism beyond next-run display v1.5.22

07 **Office, PDF, and text file icons** NEW 35

Adds Office, PDF, and text file icons to the file tree and attachment chips.

— Minor visual addition, no further detail v1.5.22

WAS THIS USEFUL? 👍 👎

Graphicle AI Graphicle

Feedback ⓘ v0.1.10 NOTES ↗

RANK

Code graph and memory for your coding agents, in one Rust program. For Claude Code, Codex, Cursor and any MCP client.

> Graphicle v0.1.10 adds an environment-variable allowlist that lets API-key model profiles reach plain-HTTP servers, friendlier guidance when the graph file is missing, and new benchmark scripts for measuring context completeness — alongside several breaking changes to sync behavior, HTML indexing, freshness handling, and the GitHub Action that require action after upgrading.

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

01 **HTTP key-host allowlist for plain-HTTP model servers** NEW 83

A model profile with an API key can now reach a plain-HTTP model server (such as a vLLM host without TLS) by listing its exact host in the new environment variable `GRAPHICLE_PROVIDER_HTTP_KEY_HOSTS`, which accepts the same format as `GRAPHICLE_PROVIDER_HTTP_HOSTS`. A host listed in `GRAPHICLE_PROVIDER_HTTP_KEY_HOSTS` is also treated as listed in the older variable.

— Names both env vars and gives a concrete server example v0.1.10

02 **HTML and HTM files stored as raw markup** BREAKING 75

A folder's `.html` and `.htm` files are now stored as raw markup instead of extracted text, so a reviewed project memory that cites one of those files goes stale after the next sync because the stored text changes from plain words to markup. Affected memories can be reviewed in the app via `graphicle open`, under Memory.

— Gives exact UI path to review affected memories v0.1.10

03 **Benchmark scripts for context completeness** NEW 73

Adds `bench/swe-bench-pro/` scripts (`e1.py`, `e1l.py` and their reports) that measure, without a model, how often a context holds every file and every line a fix changes, across 633 tasks of SWE-bench Pro's public set, comparing the pack against file-level BM25 and a repository cut to the same budget. Also adds `bench/whole-repo/lines.py`, which counts changed lines each context holds for whole-repository questions, comparing two builds question by question from each question's pull-request diff.

— Names scripts and methodology but no run command given v0.1.10

04 **Degraded prompt hook until first post-upgrade sync** BREAKING

73

The first `graphicle sync` after upgrading rewrites the saved graph and query file; until that sync runs, questions on projects of 2,000 files or more use a degraded prompt hook that returns only files matching code names in the prompt.

— Names exact threshold and remediation command v0.1.10

---

05 **GitHub Action pin update for graphicle review** BREAKING

68

In the GitHub Action for `graphicle review`, `graphicle-version` must be set to `0.1.10` and `graphicle-sha256` updated to the SHA-256 of the 0.1.10 archive for the runner, as listed in the release's `SHA256SUMS`.

— Names exact config keys and where to find the required value v0.1.10

---

06 **Guidance message when graph file is missing** IMPROVED

64

When a `graphicle graph` subcommand such as `query` or `affected` cannot find the graph file, Graphicle now prints the command to run — `graphicle-graph extract . --code-only` (the same as `graphicle graph extract . --code-only`) — instead of stopping with only a 'graph file not found' error.

— Gives the exact recovery command, but minor UX change v0.1.10

---

07 **Manual sync required for HTML with freshness off** BREAKING

60

With `freshness` set to `off`, the automatic re-sync of HTML files that an earlier version indexed does not run; users must run `graphicle sync` themselves to pick up the change.

— Names the config key and the needed command v0.1.10

└──▷ BREAKING ON UPGRADE

! The first `graphicle sync` after upgrading rewrites the saved graph and query file; until that sync runs, questions on projects of 2,000 files or more use a degraded prompt hook that returns only files matching code names in the prompt.

! A folder's `.html` and `.htm` files are now stored as raw markup instead of extracted text; a reviewed project memory that cites one of those files goes stale after the next sync, because the stored text changes from plain words to markup. Review affected memories in the app (`graphicle open`, under Memory).

! With `freshness` set to `off`, the automatic re-sync of HTML files that an earlier version indexed does not run; you must run `graphicle sync` yourself.

! In the GitHub Action for `graphicle review`, `graphicle-version` must be set to `0.1.10` and `graphicle-sha256` updated to the SHA-256 of the 0.1.10 archive for the runner (listed in the release's `SHA256SUMS`).

WAS THIS USEFUL? 👍 👎

◆ MCP Servers & Tooling

premiere-pro-mcp

Feedback ⓘ v1.27.0 NOTES ↗ RANK

Adobe Premiere Pro MCP — independent, local-first server for supported workflows. Connect Claude, Codex, or Cursor through a local CEP bridge.

> premiere-pro-mcp adds a new tool to flatten nested sequences into their parent timeline, makes normalized audio effect parameters transparent and guarded against bad writes, and extends caption style guidance with keyword highlighting.

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

01 Audio parameter scale transparency and write guard IMPROVED

91

`get_effect_properties` and `set_effect_property` now report `valueScale` and a `parameterScale` summary when an audio effect's parameters use Premiere's normalized 0-1 scale rather than real units like Hz, dB, or ms, and warn callers to calibrate by rendering. `set_effect_property` also blocks writing an out-of-range number (e.g. 80 intended as Hz) to such a normalized parameter unless `allow_out_of_range_audio_value: true` is set.

Force a raw 0-1 write to a normalized audio effect parameter when you know the value is correct and want to bypass the range guard.

```
$ set_effect_property --allow_out_of_range_audio_value true
```

— Names exact tools, fields, and the override flag with a runnable example. v1.27.0

02 Unnest sequence tool for nested clips NEW

82

`unnest_sequence_uxp` flattens a nested sequence clip into its parent timeline, preserving effects, keyframes, enabled state, and source range. Inspect mode returns the full plan before executing, and apply mode runs the flatten as one undoable transaction with per-clone readback.

— Names the tool and its two modes but no example given. v1.27.0

THINNER COVERAGE BELOW

03 Keyword highlighting in caption style guidance IMPROVED

57

`get_caption_style_guidance` now accepts a `keyword_highlight` parameter and returns per-word colour steps, along with manual guidance to use Graphics and Titles >

Upgrade Caption to Graphic.

— Names the parameter and manual UI path but no example. v1.27.0

WAS THIS USEFUL? 👍 👎

kordoc

Feedback ⓘ v4.24.28 NOTES ↗ RANK

모두 파싱해버리겠다 — HWP·HWPX·PDF·Office 문서를 Markdown으로. 양식 자동 채우기와 신구대조를 갖춘 CLI·MCP 서버 | Convert Korean documents (HWP, HWPX, PDF, Office) to Markdown — CLI and MCP server with form filling and diff

> kordoc added full memo support (reading and writing) for HWP and HWPX documents, extended HWP 5.x compatibility to form-filling and seal placement with original formatting preserved, and improved HWP 5.x text fidelity and error handling.

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

01 Memo reading and writing for HWP/HWPX NEW

95

Memos can now be read from HWP 5.x and HWPX documents into a structured `memos` result — including memo text, anchor text, anchor paragraph, character position, author, timestamp, sequence number, and containing table cell — by passing `--memos` on the CLI or `memos` in the MCP `parse_document` call; a `Memos` section is appended to the Markdown output when the flag is set. Memos can also be added without touching body text or formatting using `kordoc memo <document> --anchor "<phrase>" --text "<memo>" [--author] [-n]`, the MCP `add_memo` tool, or the library function `addMemoHwpx`; HWP 5.x files are converted to HWPX format before the memo is inserted.

Extract all memos from an HWPX document into a structured result, with a `Memos` section appended to the Markdown output — useful for auditing review comments in official documents.

```
$ npx kordoc 보고서.hwpx --memos -o 보고서.md
```

Insert a reviewer memo anchored to a specific phrase in an HWP document without altering body text or formatting — useful for programmatic annotation in a review pipeline.

```
$ npx kordoc memo 보고서.hwp --anchor "추진 일정" --text "일정 재검토 필요" --author
```

— Full command, flag and MCP tool names with runnable examples given. v4.24.28

02 HWP 5.x support in form-fill and seal commands NEW INTEGRATION

IMPROVED

85

The `fill_form` MCP tool, `fill` CLI command, `fillForm` API with `hwpx-preserve` mode, and `place_seal` MCP tool / `seal` CLI command now convert HWP 5.x to HWPX internally before processing, then output HWPX — previously MCP and library calls rejected HWP input and the CLI rebuilt the document from scratch, discarding formatting.

— Names every affected tool and mode with clear before/after. v4.24.27

THINNER COVERAGE BELOW

03 **HWP 5.x inline emphasis markers** IMPROVED 55

HWP 5.x documents now emit bold, italic, strikethrough, and underline as inline emphasis markers in the HWP parser output, consistent with HWPX output, excluding headings, table-of-contents, and filled press-field paragraphs.

— Describes scope and exclusions but no command to invoke. v4.24.28

04 **Explicit rejection of protected HWP files** IMPROVED 30

Protected or password-protected HWP files that cannot be modified are now rejected with an explicit reason, instead of silently failing.

— States the behavior change but no mechanism or surface named. v4.24.27

v4.24.26 NOTES ↗ STABLE

kordoc adds `kordoc to-hwpx` and `convert_to_hwpx` to convert HWP 5.x files to HWPX with full formatting, and improves patch editing for paragraphs with tabs or fields.

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g kordoc@4.24.26
```

git

```
$ git clone --branch v4.24.26
https://github.com/chrisryugj/kordoc.git
# already have the repo? check out this version:
$ git checkout v4.24.26
```

named in the project's README · checked 31 Oct Wrong command?

Convert HWP 5.x documents to HWPX with formatting intact using `npx kordoc to-hwpx <file.hwp> -o <file.hwpx>`; pass `--password` for encrypted documents.

> Patch editing on HWP 5.x now handles paragraphs that contain tabs or fields, modifying only the changed characters.

WAS THIS USEFUL? 👍 👎

pdf-mcp

Feedback ⓘ v3.6.0 NOTES ↗ RANK

MCP server that lets Claude Code and other AI agents read and search large PDFs, one file or a whole folder: agentic RAG with hybrid semantic + keyword search, selective page reads, tables, images, OCR, chart data, and multi-column/CJK layouts.

> pdf-mcp v3.6.0 scales corpus search to 500 PDFs with background embedding, adds on-demand force-OCR and bad-OCR detection, and tightens corpus-call reliability — while dropping Python 3.10 support and changing `pdf_corpus_warm`'s default response shape.

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

01 Background embedding control and monitoring NEW

95

By default one `pdf_corpus_warm` call is now enough; the server finishes all embeddings in the background in about 27 minutes with no further calls. Set `PDF_MCP_BACKGROUND_EMBED=0` to disable background embedding, and monitor progress via `embedding_in_background` in `pdf_corpus_warm` responses and `embeddings_pending.in_background` in `pdf_corpus_search` responses. Note the server may use CPU between calls while embedding runs in the background.

Extend the corpus cache TTL and disable background embedding on a low-CPU server where you want no between-call processing.

```
$ PDF_MCP_CACHE_TTL=172800 PDF_MCP_BACKGROUND_EMBED=0 pdf-mcp
```

— Named env var, response fields, and a runnable example v3.6.0

02 OCR quality improvements for pdf_read_pages (new) NEW

86

`pdf_read_pages` now supports `force_ocr=true` to re-OCR a page whose embedded text layer is corrupt, caching the result for `pdf_search` and returning `source='ocr'`, and (with `ocr` off) flags suspect pages with `suspect_cells` (each carrying a `clip` to render) alongside `scan_text_layer_pages` and a `hint` to detect bad OCR layers before they corrupt data.

— Exact flag, caching behavior, and response field named Named response fields but no remediation step shown

v3.6.0

03 Compact pdf_corpus_warm responses BREAKING

84

`pdf_corpus_warm` now returns a compact response by default listing only changed documents; pass `detail=true` for the full per-document list, and read `cached_docs` (and `cached_without_embeddings`) for counts of unchanged files. This changes the default shape of the response, so callers relying on a row per cached document must adapt.

— Exact flag and fields, explicit migration note v3.6.0

04 **New server configuration environment variables (new)** NEW

75

Adds `PDF_MCP_CACHE_TTL` to control how long documents stay cached, defaulting to 24 hours, with cached documents evicted when the file changes or the TTL expires, and `PDF_MCP_MAX_WORKERS` to control extraction parallelism, with values of 1 or less running extraction in-process.

Extend the corpus cache TTL and disable background embedding on a low-CPU server where you want no between-call processing.

```
$ PDF_MCP_CACHE_TTL=172800 PDF_MCP_BACKGROUND_EMBED=0 pdf-mcp
```

— Named env var and default with eviction rule Named env var and behavior, no scaling guidance

v3.6.0

05 **Corpus call deadline and retry semantics** BREAKING

75

Corpus calls may now return at their deadline with a slow document listed in `unprocessed`, or with an inline error asking to retry if the call waited out its whole budget; callers must handle both cases and repeat the call to pick up the result.

— Named response field and retry behavior, no timeout value given v3.6.0

06 **Corpus search scaled to 500 PDFs** IMPROVED

73

`pdf_corpus_warm`, `pdf_corpus_overview`, and `pdf_corpus_search` now rank across up to 500 PDFs in one call, up from 100, so large folders no longer need to be split across calls with incomparable scores.

— Named endpoints and before/after numbers, no usage example v3.6.0

THINNER COVERAGE BELOW

07 **Python 3.11 minimum requirement** BREAKING

40

pdf-mcp now requires Python 3.11 or later; Python 3.10 is no longer supported.

— Clear requirement but no migration detail v3.6.0

└──▷ BREAKING ON UPGRADE

- ! `pdf_corpus_warm` no longer returns a row for each already-cached document; read `cached_docs` for the count, or pass `detail=true` for the full list.
- ! Python 3.10 is no longer supported; pdf-mcp now requires Python 3.11 or later.
- ! Corpus calls may now return at their deadline with a slow document in `unprocessed`, or with an inline error asking to retry if the call waited out its whole budget — callers must handle both cases and repeat the call to pick up the result.
- ! The server may use CPU between calls due to background embedding; set `PDF_MCP_BACKGROUND_EMBED=0` to disable this behaviour.

WAS THIS USEFUL?  

godot-mcp

Feedback ⓘ v0.34.0 NOTES ↗ RANK

Godot-MCP — Model Context Protocol (MCP) integration for the Godot Engine. dev.

> Godot-MCP added a CLI `configure` command for hiding and showing MCP tools, and shipped v0.34.0 with a major RQL update (bulk create, vector arithmetic, named arguments), tighter Code-Exec limits and timeouts, and a breaking rename of the plugin dock's tool visibility property.

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

01 **CLI command to hide or show MCP tools** NEW 85

The `configure [path]` command uses `--enable-*` or `--disable-*` flags to control which MCP tools, prompts, and resources clients see listed, writing visibility state to `.godot-mcp/features.json`. A hidden tool still executes when called by name.

Hide a specific MCP tool from client listings while keeping it callable by name, then verify the change in the project-local features file.

```
$ godot-mcp configure --disable-<tool-name>
```

Re-enable a previously hidden tool so MCP clients can see and list it again.

```
$ godot-mcp configure --enable-<tool-name>
```

— Exact commands and config file path given product docs

02 **RQL bulk operations, vector math and named arguments** IMPROVED 75

RQL now supports bulk create operations, vector arithmetic, named arguments (including agents calling methods via RQL's named-args support), side-effect-free overloads, and compact receipts. Typed arrays of resources accept literals and refs, trigonometric and `$count` expressions are available, validators are now called for node and file calls (dropping earlier tuple workarounds), and `AddSource` / `AddSource<T>` support declared source types with Godot's own creation examples and `$ref` node arguments.

— Many named additions but no runnable usage example v0.34.0

03 **openProject CLI command reports process ID** IMPROVED 60

The `openProject` CLI command now returns a PID promise from `spawnImpl` so the process ID is always reported.

— Named command and internal function, no example call v0.34.0

THINNER COVERAGE BELOW

04 **Code-Exec output limits and script-execute timeout** IMPROVED

55

Code-Exec now enforces its own output limits and adds a `script-execute` timeout and execution guards, replacing the plugin's earlier log caps.

— Names the timeout but gives no example invocation v0.34.0

05 **Reload-gate signals in-progress C# reload** IMPROVED

50

Reload-gate now returns a `NotExecuted` result when a C# reload is in progress rather than silently dropping calls.

— Named result value but no example of handling it v0.34.0

06 **Plugin dock tool property renamed to Visible** BREAKING

50

The tool `Enabled` property is renamed to `Visible` in the plugin dock; any automation or config referencing the old name must be updated.

— Clear rename stated but no migration steps v0.34.0

07 **Extension commands now Python-only** BREAKING

30

Extension commands are now Python-only, dropping the PowerShell scripts.

— No migration detail or affected command list given v0.34.0

└──▷ **BREAKING ON UPGRADE**

! The tool `Enabled` property is renamed to `Visible` — any automation or config referencing the old name must be updated.

WAS THIS USEFUL?  

slack-skills-plugin

Feedback ⓘ v1.6.0 NOTES ↗

Claude Code, Codex, Cursor, and npx skills plugin that enables your AI tools with a Slack MCP Server and Slack Developer Skills.

slack-skills-plugin adds `--app`, `--template`, and `--team` arguments to the agent setup prompt to link new projects to apps already created in App Settings

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.6.0 https://github.com/slackapi/slack-skills-plugin.git
# already have the repo? check out this version:
$ git checkout v1.6.0
```

Set up a project for an app already created in App Settings by passing `--app`, `--template`, and `--team` to the agent setup prompt, which skips workspace, language, and template selection and runs `slack create --app` to link the project to the existing app.

The `create-slack-app` skill now uses `slack-samples/bolt-python-blank-template` when a developer picks the App Settings blank template for Python, instead of the Python starter template.

WAS THIS USEFUL? 👍 👎

concord-mcp

Feedback ⓘ v0.19.0 NOTES ↗ RANK

Live messaging for coding agents

> concord-mcp's Gemini adapter now starts its inbox monitor automatically, removing a manual setup step.

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

01 **Automatic inbox monitor startup for Gemini adapter** IMPROVED 60

The `concord adapters gemini` CLI command now starts the inbox monitor automatically on the first conversation turn, removing the need to start it manually.

– Names exact CLI command and behaviour change, no deeper mechanism given. v0.19.0

WAS THIS USEFUL? 👍 👎

mcp-server

Feedback ⓘ v2.4.5 NOTES ↗ RANK

BrowserStack's Official MCP Server

> BrowserStack's MCP server adds Kiro support and switches screenshot delivery to native MCP image blocks.

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

01 **Screenshots sent as MCP image blocks** IMPROVED 45

Screenshots are now sent as MCP image blocks instead of base64 text, making visual output directly usable in MCP-aware clients.

– Explains the before/after change but no API or config detail v2.4.5

02 **BrowserStack Kiro support added** NEW 25

The MCP server now adds BrowserStack Kiro power, enabling Kiro-based capabilities alongside existing BrowserStack integrations.

– Names the integration but no mechanism or usage detail v2.4.5

WAS THIS USEFUL? 👍 👎

agentcad

Feedback ⓘ v0.7.1 NOTES ↗ RANK

CAD CLI and MCP server for AI agents

> AgentCAD v0.7.1 adds a part-linked review comment system to the live project viewer, ships a cloud CLI with a local MCP adapter for configured hosts, and strengthens STEP assembly inspection with stable geometry IDs and image-linked imports.

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

01 **Part-linked review comments in live viewer** NEW 95

Reviewers can add spatial comments to the live project viewer with `agentcad review list --status open`, `agentcad review comment`, `agentcad review reply`, and `agentcad review resolve`; comments are stored as plain JSON under `.agentcad/reviews` and become available to the agent on its next turn. Comments can be pinned to named parts and revision scopes using `--part <name>`, `--version current|previous|both`, `--scope previous|both`, and `--point-mm X,Y,Z` for precise 3D placement.

Pin a review comment to a specific named part and surface point so the agent knows exactly where to focus on its next turn.

```
$ agentcad review comment --message "The rib here looks too
thick" --part support_rib --version current --point-mm 12.5,0,8.2
```

Read all open review threads from the agent side before starting a revision, then reply with the change made.

```
$ agentcad review list --status open
agentcad review reply C1 --message "Reduced rib thickness to 2 mm
in current revision." --version current
agentcad review resolve C1 --message "Implemented in the current
revision." --version current
```

— Full command set, flags and storage path all given v0.7.1

02 **Cloud CLI and local MCP adapter** NEW 60

A new cloud CLI and local MCP adapter is available for configured hosts, installed by upgrading to `agentcad[mcp]==0.7.1` via `python -m pip install --upgrade 'agentcad[mcp]==0.7.1'`.

Install MCP support and upgrade to v0.7.1 to enable the cloud CLI and local MCP adapter for configured hosts.

```
$ agentcad viewer stop
python -m pip install --upgrade "agentcad[mcp]==0.7.1"
```

— Install command given but adapter's behaviour unexplained v0.7.1

THINNER COVERAGE BELOW

03 **STEP assembly inspection with geometry IDs** IMPROVED

45

STEP assembly intake now supports inspection with stable geometry IDs, ID-linked images, ranked samples, and default inspection bundles for both imported and generated models. Local MCP imports now also include these inspection images alongside the model data.

— Concepts named but no commands or flags to act on v0.7.1

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Inference Serving & Optimization

Niko1221 Strata

Feedback ⓘ v0.2.9 NOTES ↗ RANK

Strata is a local inference engine with one-click install for Windows and Linux, providing an OpenAI/Anthropic-compatible API on localhost for models like Qwen and DeepSeek.

> Strata's v0.2.9 release adds support for new DeepSeek model formats and quantizations, turns on multi-GPU use by default for both AMD and Intel Arc B-series cards, and introduces a string of hardware-specific performance defaults (parallel request batching on Linux/NVIDIA, DeltaNet kernel tuning, Strix Halo matrix-core kernels) alongside new diagnostic switches.

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

01 New DeepSeek model sets and GGUF shard layouts NEW

85

Adds Unsloth's 1-bit DeepSeek sets `UD-IQ1_S` and `UD-IQ1_M` as selectable options in setup and via `strata launch` on Windows, Linux (NVIDIA and AMD), DGX Spark, and Mac. Adds support for the ISTA four-shard DeepSeek GGUF layout and ISTA's six-shard layout through Use a local GGUF in the app and via `--model-path`, refusing a cut or missing shard with a message naming the missing part.

— Names exact model sets, shard layouts and the --model-path flag v0.2.9

02 DeepSeek defaults to all AMD GPUs BREAKING

80

DeepSeek on multiple AMD cards now uses all available cards by default with `strata launch` and no `--gpu`. Specify a single card with `strata launch --gpu N`, a choice which is now remembered. Breaking: an install previously set up on one AMD card will move to all available cards at next start without rebuilding; run `strata launch --gpu N` to stay on one card.

Pin DeepSeek to a single AMD GPU to override the new all-cards default, and have that choice remembered.

```
$ strata launch --gpu 0
```

— Exact flag and runnable example for the behavior change v0.2.9

03 Intel Arc B-series multi-card support and defaults NEW

80

Two or more Intel Arc Pro B-series cards (B65, B60, B50) are now recognized by `strata launch`, filling both cards automatically when no `--gpu` is given. These cards now run

prompt attention on the int8 path by default (disable with `STRATA_PROMPT_ATTN_ESIMD=0`), and two B-series cards with 8 or more slots now run 2 pipelined groups by default (disable with `--batch-groups 1` or `STRATA_BATCH_GROUPS_AUTO=0`).

— Names hardware models and three concrete disable flags v0.2.9

---

04 **Default parallel requests on Linux/NVIDIA** IMPROVED

80

Parallel requests on Linux with NVIDIA cards are now on by default when `--parallel` or `--batch` of 2 or more is set, with nine qualifying settings configured automatically by the server: `STRATA_CKPT_POOL`, `STRATA_CKPT_POOL_READY`, `STRATA_SLOT_CKPT_MOVE`, `STRATA_SLOT_COPY_DIRECT`, `STRATA_SLOT_COPY_BATCHSYNC`, `STRATA_BATCH_GDN_GROUPS`, `STRATA_BATCH_ATTN_ROWS`, `STRATA_BATCH_SELF_COMMIT`, `STRATA_QSA_SLOTS`.

Roll back the engine to 0.1.54 if the new Linux parallel or kernel defaults cause regressions.

```
$ strata components rollback
```

— Lists all nine auto-configured settings and trigger condition v0.2.9

---

05 **Gated DeltaNet kernel defaults for NVIDIA and Radeon** IMPROVED

70

Gated DeltaNet prompt kernels now run on tensor cores by default on Linux with NVIDIA cards (Qwen3.8), and the chunked kernel now starts at 64 tokens by default on Linux with all-gfx12 Radeon cards. Disable with `STRATA_GDN_TC=0` and `STRATA_GDN_CONVL2=0` (NVIDIA) or `STRATA_GDN_CHUNKED=0` (Radeon gfx12).

— Names exact env flags and hardware/model conditions v0.2.9

---

06 **Strix Halo F16 matrix-core prompt kernels** IMPROVED

70

Strix Halo on Windows (gfx1151) now uses F16 matrix-core prompt kernels by default (`STRATA_PF_F16`), with `STRATA_SELECT_WMMA` (block scorer on matrix cores) also on by default. Turn either off with `NAME=0`, for example `STRATA_PF_F16=0`.

— Names the exact flags and how to disable them v0.2.9

---

07 **Parallel-work diagnostic switches and replica profile** NEW

60

Research switches `STRATA_PAR_COUNTERS`, `STRATA_COALESCE_PREFILL`, and `STRATA_CRASH_BT` are now available for parallel-work diagnostics, off by default, as is

the replica profile via `setup --replica-profile`.

— Names exact switches but gives little on their behavior v0.2.9

└──▷ **BREAKING ON UPGRADE**

! DeepSeek on multiple AMD cards: an install previously set up on one AMD card (setup's automatic pick) will move to all available cards at next start without rebuilding. To stay on one card, run `strata launch --gpu N`.

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

KoboldCpp

Feedback ⓘ v1.124 NOTES ↗ CODE ↗ RANK

KoboldCpp runs GGUF language models locally as a single-file application with a built-in web UI and API.

> KoboldCpp's v1.124 release adds audio.cpp-powered music and speech generation, new performance-tuning flags for parallel context pooling and RPC/MoE caching, and broadens support across AI Horde, image generation, Kobold Lite, and base model architectures.

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

01 Parallel context pooling and cache-sizing flags NEW

92

New CLI flags give finer control over memory and loading: `--parallelcontext` sets the total KV context pool shared by concurrent requests when used with `--parallelrequests` (defaults to `contextsize` if unset); `--rpccache` caches RPC host tensors to a directory; `--lazy-mode` exposes upstream lazy model loading; and `--moe-cache-mib` sets the MoE expert cache size in MiB.

Run many large concurrent requests without them queuing by giving the shared KV pool more room than the per-request context alone would provide.

```
$ koboldcpp --model model.gguf --parallelrequests 4 --contextsize 8192 --parallelcontext 32768
```

Offload RPC tensors to a local cache directory so repeated loads skip re-transferring weights over the network.

```
$ koboldcpp --model model.gguf --rpccache /mnt/rpc-cache
```

Reduce VRAM pressure on a large MoE model by capping the MoE expert cache and deferring weight loads with lazy mode.

```
$ koboldcpp --model mixtral.gguf --lazy-mode --moe-cache-mib 4096
```

— Named flags with defaults and runnable example commands for each. v1.124

02 audio.cpp music and TTS model support NEW

68

KoboldCpp integrates audio.cpp, adding YuE2 music generation via three templates — `YuE2-No-LLM` (lyrics only), `YuE2-Small-LLM` (4B, recommended), and `YuE2-Large-LLM` (large GPU) — plus 17 ported TTS models: `BreezeTTS 2`, `Ming 3.0 TTS`, Qwen3-TTS Base, `Qwen3-TTS VoiceDesign`, `VoxCPM2`, Chatterbox, Chatterbox Turbo, `CosyVoice 3`, Fish Audio S2 Pro, Kitten TTS 2, Maya1, `NeuTTS 2E`, `OmniVoice`, `PocketTTS`, `Soprano 1.1 80M`, Supertonic 3, and `VibeVoice 1.5B`, several of which support voice cloning and instruction-driven synthesis.

— All model names listed but no invocation command given. v1.124

---

03 **New image generation models and tokenizer handling** NEW

68

Image generation gains support for MOSS Image, Fibo 2 Edit, and Flux 3 Klein models via an `sd.cpp` sync, and can now use tokenizers baked into the text encoder LLM, while an explicit `tokenizer.json` is still accepted in the `clip2` slot.

— Named models and config slot but no runnable example. v1.124

---

04 **Chat Completions for AI Horde workers** NEW

62

AI Horde workers now support Chat Completions automatically on upgrade, enabling jinja templating, reasoning, tool calls, streaming, and vision over AI Horde; activated by toggling chat completions mode in `https://lite.koboldai.net`.

— Lists enabled capabilities and toggle location but no API detail. v1.124

THINNER COVERAGE BELOW

---

05 **Kobold Lite responses endpoint and tool/search improvements**

IMPROVED

57

Kobold Lite adds `/v1/responses` endpoint support alongside SSE parser improvements and improvements to tool call and web search handling.

— Names endpoint but improvements described only in brief. v1.124

---

06 **Transcribe tab in MusicUI** NEW

55

MusicUI at `http://localhost:5001/musicui` adds a transcribe tab that uses Whisper models to generate reference transcripts for TTS models that require them.

— Exact UI path given but no further mechanism detail. v1.124

---

07 **Agent MCP wizard and paste mode** NEW

55

KoboldCpp Agent adds a `/mcp` wizard for setting up custom MCP endpoints and a `/paste` mode for handling multiline text input.

— Named commands but no further explanation of behavior. v1.124

---

08 **Animated WebP default for video preview** IMPROVED

50

Animated WebP is now the default preview format for video generation; GIF remains supported via SDUI settings, and AVI remains the default download format.

— Describes format defaults but gives no setting path or command. v1.124

09 **New supported model architectures** NEW 35

KoboldCpp adds support for Nemotron 3.5 Bolt 74B-A9B, Mistral Large 4, LFM3, and Muse Spark model architectures from upstream.

— Bare list of newly supported architectures, no further detail. v1.124

WAS THIS USEFUL? 👍 👎

llamastash Feedback ⓘ v0.10.1 NOTES ↗

Zero-overhead, terminal-native local-LLM orchestrator. Run any engine, any model via the same proxy.

llamastash adds a `preset` label to `/metrics` launch_info, MoE expert-spill placement for oversized models on discrete GPUs, and an MLX recipe for Apple Silicon.

└──▷ GET THIS VERSION ▾

Cargo

```
$ cargo install llamastash --version 0.10.1
# already installed? upgrade to this version:
$ cargo install llamastash --version 0.10.1 --force
```

git

```
$ git clone --branch v0.10.1
https://github.com/llamastash/llamastash.git
# already have the repo? check out this version:
$ git checkout v0.10.1
```

named in the project's README · checked 31 Oct Wrong command?

The `/metrics` endpoint now includes a `preset` label on `llamastash:launch_info` so a scrape can tie each launch's request and token counters back to the preset it started from; launches with no preset print the label empty, and the JSON `launches` rows carry the same fact as a `preset` field (`null` when none).

The recommender now considers MoE models whose weights exceed VRAM on a discrete GPU, using llama.cpp's `--fit` to keep the active slice on the GPU and placing experts in system RAM within a budget of total RAM minus a 15 % reserve, clamped to 4-32 GiB; picks that spill show `· experts in RAM ~NN%` and rank lower by spill fraction.

NEW SUPPORTED PLATFORM `RECIPES.md` now includes an MLX recipe covering mlx-lm and oMLX as generic entries on Apple Silicon, where `llamastash pull mlx-community/<repo>` feeds either engine and each mints its rows through a `rows_from` lister script.

NEW INTEGRATION Artificial Analysis Intelligence Index benchmark scores

UPDATED INTEGRATION whichllm discovery tracker updated to 0.5.23

NEW INTEGRATION Curated open-weight model catalog in data/frontier-models.yaml

WAS THIS USEFUL?  

◆ Agentic Infrastructure

sandboxd

Feedback ⓘ v0.3.25 NOTES ↗ RANK

Open-source, self-hosted AI app builder — an agent builds real apps in isolated sandboxes on your own server, each live at a preview URL. Self-host in one command.

> sandboxd adds an AI agent (Hermes) that can build apps via chat or Telegram, automatically picks a framework stack from your project idea, and now applies sandbox.yaml changes to a running preview without a restart — while trimming several options from the home screen.

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

01 Live framework switching via sandbox.yaml IMPROVED

88

Editing `sandbox.yaml` now takes effect immediately: `runtimed` applies the new start command, health check, and build check to the running preview without waiting for a workspace restart. It reverts changes that would break the preview — an invalid file, a port conflict, or a web-app-to-worker switch — while keeping the app running.

— Names config file, daemon, and exact revert conditions v0.3.24

02 Automatic framework stack selection from project idea NEW

73

sandboxd now automatically picks a framework stack — Next.js, Express, FastAPI, or a background worker — when the project idea names them, defaulting to React/Vite otherwise, and shows its reasoning under the prompt. If the idea names an unsupported framework (Vue, Svelte, Astro, Flask, Django) or unsupported language (PHP, Ruby, Java, .NET, Go, Rust, native mobile), it surfaces a plain warning before any build starts.

— Lists exact supported and unsupported stacks, behavior described v0.3.24

03 Hermes AI agent for building apps NEW

70

A new Agents page lets you add an AI agent — starting with Nous Research's Hermes — to build and manage apps via chat within sandboxd or over Telegram. It requires an OpenAI-compatible model connection, is capped at 10 new apps per day, cannot delete apps, and logs every action it takes.

— Names mechanism, limits, and requirement but no exact commands v0.3.25

THINNER COVERAGE BELOW

04 Home screen options removed BREAKING

50

The App library, starter templates, Import from Git, and Blank project options have been removed from the home screen UI, though the engine API still supports Git import and creating empty apps.

— States what was removed but no migration path given `v0.3.24`

└──▷ **BREAKING ON UPGRADE**

! The App library, starter templates, Import from Git, and Blank project options have been removed from the home screen UI (the engine API still supports Git import and empty apps).

WAS THIS USEFUL? 👍 👎

OpenAI TypeScript SDK

Feedback ⓘ v7.46.0 NOTES ↗

RANK

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

> OpenAI's TypeScript SDK adds new agent capabilities including session review, metadata filtering, and experimental time awareness, alongside Realtime API improvements (SIP DTMF, typed references), image edit markup overlays, and early support for organization external key management.

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

01 **Realtime API: SIP DTMF, typed references and WebSocket reliability**

NEW 50

The Realtime API adds `SIP DTMF` commands and delivery events, typed Realtime references, and improved WebSocket delivery reliability.

— Names three concrete Realtime additions, no exact endpoints v7.46.0

02 **Agent session filtering, review and tool settings** NEW

45

Agent sessions can now be filtered by metadata when listing sessions, and the SDK adds session review and tool settings for inspecting and configuring agent sessions.

— Names the capabilities but no exact API or flag given v7.46.0

03 **Experimental agent time awareness** NEW 30

OpenAI TypeScript SDK adds experimental agent time awareness so agents can reason about the current date and time.

— Describes purpose but no mechanism or config shown v7.46.0

04 **Hosted tool usage in Responses API** IMPROVED 30

The SDK now exposes hosted tool usage in Responses API responses.

— States the change but no field or schema named v7.46.0

05 **Image edit markup overlays** NEW 30

OpenAI TypeScript SDK adds image edit markup overlays for annotating image edit inputs.

— Describes feature but no API or parameter named v7.46.0

06 **Latest text format parsing in Responses API** IMPROVED

15

OpenAI TypeScript SDK now parses Responses using the latest text format update.

— Vague reference to a format update with no specifics v7.46.0

07 **Agent context compaction** NEW

10

OpenAI TypeScript SDK adds agent context compaction.

— Only named in summary, no detail provided v7.46.0

08 **Organization external key management** NEW

10

OpenAI TypeScript SDK adds organization external key management.

— Only named in summary, no further detail given v7.46.0

WAS THIS USEFUL? 👍 👎

OpenAI Python SDK

Feedback ⓘ v3.42.0 NOTES ↗ RANK

The OpenAI Python SDK provides typed Python clients for OpenAI model APIs, streaming, and asynchronous requests.

> OpenAI's Python SDK v3.42.0 adds new agent capabilities—context compaction, session review, and time awareness—alongside per-request API key scoping, telephony SIP DTMF support, image edit overlays, external key management, and realtime WebSocket reliability improvements.

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

01 **Agent context compaction, session review, and time awareness**

NEW 50

Adds context compaction settings to control how agent conversation context is trimmed, session review and tool settings for inspecting and configuring agent sessions, and experimental time awareness so agents can reason about the current date and time.

– Names three agent additions but no concrete API calls or flags. v3.42.0

02 **Realtime WebSocket reliability improvements** IMPROVED

45

Adds WebSocket warmup support and preserves cleanup cancellation for more reliable realtime connections, and improves WebSocket recovery and delivery reporting for realtime sessions.

– Groups related realtime fixes but lacks specific API names. v3.42.0

03 **Per-request API key scoping** NEW 40

Callable API keys can now be scoped to each individual request so that per-request credentials are isolated rather than shared across the client.

– Describes behavior but no code example or parameter name. v3.42.0

04 **SIP DTMF commands for telephony** NEW 40

Adds SIP DTMF commands and delivery events for telephony integrations.

– Names the feature area but not specific commands or endpoints. v3.42.0

05 **Response format and local output types in responses API**

IMPROVED 40

Supports response format updates and local output types in the responses API.

– Names the responses API but not specific fields or formats. v3.42.0

06 **Organization external key management** NEW 35

Organizations can manage encryption keys outside of OpenAI via new external key management support.

— States purpose but no config key or endpoint named. v3.42.0

07 **Markup overlays in image edit API** NEW 25

Image edit markup overlays added to the image editing API.

— Bare mention with no mechanism or usage detail. v3.42.0

WAS THIS USEFUL? 👍 👎

AWS Amazon Bedrock AgentCore

Feedback ⓘ

Gateway: MCP task support for long-running tool calls (2026-10-30) BLOG ↗

RANK

Amazon Bedrock AgentCore provides managed agent runtimes, memory, identity, gateways, observability, and evaluation services.

> Amazon Bedrock AgentCore's Gateway now supports asynchronous long-running tool calls via the MCP Tasks extension, while Runtime gained a higher environment variable ceiling for V2 agents and new session endpoints for connecting into running agent sessions.

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

01 MCP Tasks extension for long-running Gateway tool calls NEW

88

Amazon Bedrock AgentCore Gateway now supports the MCP Tasks extension (`io.modelcontextprotocol/tasks`) on MCP protocol version `2026-07-28` , letting a tool on an MCP server target return a task ID immediately instead of holding the connection open for long-running operations such as batch jobs, CI pipelines, or report compilation. Callers poll task status and retrieve the final result via `tasks/get` , send mid-task input with `tasks/update` , and cancel a running task with `tasks/cancel` ; the gateway issues its own task ID and routes each follow-up from that ID. On gateways using JWT or IAM inbound authorization, each task is bound to the caller that created it, and tasks run without requiring MCP sessions or response streaming.

— Names protocol version and exact method names tasks/get, tasks/update, tasks/cancel

Gateway: MCP task support for long-runn…

02 Higher environment variable limit for V2 agents IMPROVED

60

V2 agents in Amazon Bedrock AgentCore Runtime now support up to 4 KB of environment variables, up from 1.5 KB for direct code deployments and 2.5 KB for container agents, matching the V1 platform limit.

— Gives exact before/after limits but no config mechanism to act on

Runtime: Increased environment variable…

THINNER COVERAGE BELOW

03 Session endpoints for Runtime agent sessions NEW

25

Amazon Bedrock AgentCore Runtime now supports session endpoints, letting you connect directly to applications running inside an agent's session.

— Bare description with no endpoint name, path, or usage detail

Runtime: Session endpoints (2026-10-20)

WAS THIS USEFUL? 👍 👎

skillhub

Feedback ⓘ v0.2.25 NOTES ↗ RANK

Self-hosted, open-source agent skill registry for enterprises. Publish & version skill packages, govern with RBAC and audit logs, deploy on-premise with Docker or Kubernetes.

> SkillHub's v0.2.25 release adds permission-aware Suite membership hints on skill cards, Korean language support for the Web client, and passive semantic attributes for optional frontend analytics integrations.

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

01 **Viewer-safe Suite membership hints on skill cards** NEW

42

Skill cards now show Suite membership hints limited to Suites the current viewer can access, while preserving individual skill navigation and installation.

— Explains behaviour and permission scoping but no UI path or config given v0.2.25

02 **Passive semantic attributes for analytics integrations** NEW

38

SkillHub adds passive semantic attributes for frontend analytics integrations, with optional integration examples; analytics collection remains disabled by default and no tracking SDK is loaded or events sent automatically.

— States default-off behaviour but no concrete attribute names or example given v0.2.25

03 **Korean language support in Web client** NEW

30

SkillHub adds Korean translations to the Web client and related-Suite badges, with browser-language detection and a persistent language preference.

— Describes feature but no setting name or toggle to act on v0.2.25

WAS THIS USEFUL? 👍 👎

Snowflake Cortex AI

Feedback

Oct 30, 2026: Cortex Training (Preview)

NOTES ↗

Snowflake Cortex AI provides managed language-model functions, search, and agents that work with data governed in Snowflake.

Snowflake Cortex AI adds Cortex Training, a serverless fine-tuning service for open-weight LLMs with an OpenAI-compatible inference endpoint

Deploy a base model or fine-tuned checkpoint to an OpenAI-compatible endpoint and stream chat completions from it, with no GPU cluster provisioning required.

Run supervised fine-tuning (SFT) and reinforcement learning (RL) with LoRA or full-parameter training via the Python client or open-source training frameworks.

Save and resume from training checkpoints, and run inference against any saved checkpoint.

Cortex Training is billed in AI Credits metered per second for on-demand GPU usage, and is currently available only in commercial AWS regions.

WAS THIS USEFUL? 👍 👎

◆ Inference Providers & GPU Clouds

Eigen Labs Darkbloom

Feedback ⓘ v0.10.5 NOTES ↗

CODE ↗ RANK

Darkbloom is a for the innovators pushing the frontier.

> Darkbloom's v0.10.5 release adds a new `darkbloom sandbox` CLI for managing Sandbox CI jobs and delivers a cluster of performance improvements spanning SSD caching, hash verification, token latency, write-ledger accounting, and reconnection handling.

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

01 **Sandbox CI job management CLI** NEW 80

A new `darkbloom sandbox` command lets operators set up, monitor, pause, and remove Darkbloom Sandbox CI jobs directly from the provider CLI. Status can be queried in machine-readable form with `darkbloom sandbox status --json`, and jobs can be paused and resumed with `darkbloom sandbox pause` and `darkbloom sandbox resume` — for example before a planned maintenance window.

Get machine-readable add-on status to feed into monitoring scripts or CI pipelines.

```
$ darkbloom sandbox status --json
```

Temporarily stop accepting sandbox jobs — for example, before a planned maintenance window — then resume when ready.

```
$ darkbloom sandbox pause
# ... maintenance ...
darkbloom sandbox resume
```

— Named subcommands and a runnable flag make this immediately usable. v0.10.5

THINNER COVERAGE BELOW

02 **SSD cache maintenance and lookup performance** IMPROVED 55

SSD cache maintenance now rereads only new or changed headers, making root-maintenance passes 11× faster. SSD cache lookup stalls drop from ~1s to under 28ms by retiring sweeps in short slices.

— Clear before/after numbers but no user-facing control to act on. v0.10.5

03 **Faster follow-up token latency for Qwen conversations**

IMPROVED 50

Follow-up time-to-first-token for dense Qwen conversations drops by 16.5% by restoring from just below the prompt end.

— Specific model, metric and mechanism, but nothing to configure. v0.10.5

04 **Faster full-hash verification via overlapped reads** IMPROVED

40

Uncached weight reads now overlap with SHA-256 hashing, making full-hash verification 1.6× faster.

— Mechanism and figure given, but no actionable surface. v0.10.5

05 **Batched SSD write-ledger charging for attention donations**

IMPROVED 40

SSD write-ledger charges now happen once per attention donation, making donations ~5× faster.

— Mechanism and speedup figure named, no actionable surface. v0.10.5

06 **Faster provider reconnection** IMPROVED 20

The provider now reconnects faster after a long session or a 1012 close.

— Bare claim with no mechanism or numbers. v0.10.5

WAS THIS USEFUL?  

Baseten

Feedback ⓘ Mixed-mode rolling promotions NOTES ↗ RANK

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

> Baseten's rolling deployment promotions now support mixed-mode configuration, combining surge and unavailability controls in a single rollout.

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

01 **Mixed-mode rolling promotions** IMPROVED 55

Rolling promotions now support setting max surge and max unavailable simultaneously, allowing control over how many extra replicas can be provisioned while also capping how much serving capacity can temporarily decrease during the same rollout.

— Explains mechanism but no config key or flag names given Mixed-mode rolling promotions

WAS THIS USEFUL? 👍 👎

fal

Feedback ⓘ Fallback Chains NOTES ↗ RANK

Provides hosted generative-media model APIs and a serverless GPU platform for deploying custom inference workloads.

> fal introduced fallback chains, letting a single API request try up to three models in sequence when earlier ones fail.

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

01 **Fallback chains for model execution** NEW 87

A single request to `POST https://fal.run/execute` can specify a primary model plus up to two fallback models; fal automatically runs the next model if the primary fails with an allowed error, returning the first successful result along with a record of each attempt. Each model in the chain can define its own input, start timeout, and list of allowed errors that trigger the next fallback.

— Names exact endpoint, method, and per-model config options. Fallback Chains

WAS THIS USEFUL? 👍 👎

RunPod

Feedback ⓘ announced · Oct 28, 2026 BLOG ↗ RANK

RunPod is an AI developer cloud platform for experimenting, training, fine-tuning, deploying, and scaling AI workloads across 31 global regions.

> RunPod launched Batch Inference as a generally available capability for submitting large-scale inference jobs on its platform.

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

01 **Batch Inference now generally available** NEW 20

Batch Inference is now generally available on RunPod, allowing users to submit large-scale inference jobs on the platform.

— No mechanism, limits, or usage details given beyond GA status. launch-20261028-3a22ae49

WAS THIS USEFUL? 👍 👎

◆ AI / LLM Gateways

ai-gateway

Feedback ⓘ v1.6.0 NOTES ↗ RANK

AI Gateway routes AI model requests through a self-hosted gateway with observability and security controls.

> ai-gateway v1.6.0 introduces versioned, namespace-scoped cache fingerprinting for completion caches and adds configurable HTTP client construction for MCP clients.

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

01 **Versioned cache fingerprinting for completion caches** BREAKING

88

Cache keys for exact-match completion caches are now created via `pkg/cache.Fingerprint`, using the response-cache plugin's encoder, and are versioned and namespace-scoped with a `v1:` prefix and opaque namespace. This is a breaking change: existing in-memory cache entries will miss after the encoding change (no storage migration required), and `RoutingMetadata`, `StreamOptions`, and `ClientStreamOptions` are now excluded from fingerprints, so applications must scope caches by routing configuration and handle stream framing and usage options separately.

— Names exact function, prefix and excluded fields, plus migration impact v1.6.0

02 **Configurable MCP HTTP client construction** NEW

75

MCP clients can now be constructed with a caller-supplied HTTP client, a response-size limit, and an initialization identity using `mcp.NewClientWithOptions`; the existing `mcp.NewClient` signature and defaults remain unchanged.

— Names exact new function and parameters, no usage example given v1.6.0

└──▷ BREAKING ON UPGRADE

! Cache fingerprints now use a `v1:` prefix and an opaque namespace; existing in-memory cache entries will miss after the encoding change. No storage migration is required.

! `RoutingMetadata`, `StreamOptions`, and `ClientStreamOptions` are excluded from fingerprints; applications must scope caches by routing configuration and handle stream framing and usage options separately.

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

redis-vl-python

Feedback ⓘ v0.29.0 NOTES ↗ RANK

Redis Vector Library. The AI-native Python client for Redis.

> redis-vl-python v0.29.0 adds debug logging to the query path and changes error behavior for partial route deletions on Redis Cluster.

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

01 **PartialDeletionError on cluster route deletion** BREAKING

75

On Redis Cluster, `SemanticRouter.remove_route` and `delete_route_references` now raise `PartialDeletionError` instead of returning normally when some reference keys cannot be deleted. Code that reads `to_dict()["routes"]` by position should instead look routes up by `name`.

— Names exact exception, methods, and required migration step v0.29.0

THINNER COVERAGE BELOW

02 **Debug logging in query path** NEW

55

Every query() call in the `index` module now debug-logs the query type, latency, and result count.

— Names the module and logged fields but no log format or flag v0.29.0

└──▷ BREAKING ON UPGRADE

! On Redis Cluster, `SemanticRouter.remove_route` and `delete_route_references` now raise `PartialDeletionError` instead of returning normally when some reference keys cannot be deleted. Code that reads `to_dict()["routes"]` by position should look routes up by `name`.

WAS THIS USEFUL? 👍 👎

uteke

Feedback ⓘ v0.22.1 NOTES ↗ RANK

The Brain for Your AI — Local-first memory engine for AI agents. Store, recall, and search memories with semantic embeddings.

> uteke v0.22.1 hardens its self-update mechanism with cosign-based signature verification and removes the older minisign scheme, while also extending update-check opt-outs to the server and MCP binaries and adding a command to link legacy room records.

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

01 Minisign verification removed; reinstall required

BREAKING

83

The minisign signature of `checksums-sha256.txt`, `scripts/sign-release.sh`, `scripts/uteke-release.pub`, and the `minisign-verify` dependency are removed in favor of cosign. Builds before v0.22.1 cannot verify a keyless cosign-signed release and must be reinstalled once (`brew upgrade codecoradev/tap/uteke`, `cargo install uteke-cli`, or a manual download) before `uteke upgrade` will work again, since it now refuses any release older than v0.22.1.

— Names exact files removed and gives concrete migration commands. v0.22.1

02 Update-check opt-out extended to serve and MCP binaries

IMPROVED

82

The daily GitHub update check can now be disabled in `uteke-serve` and `uteke-mcp`, not just the CLI, by setting `UTEKE_NO_UPDATE_CHECK` (any value) or `update_check = false` in the global `uteke.toml`.

Prevent `uteke-serve` from phoning GitHub on every daily update check in an air-gapped or privacy-sensitive deployment.

```
$ UTEKE_NO_UPDATE_CHECK=1 uteke-serve --port 8767
```

Permanently disable the update check for all three binaries via the global config file.

```toml
update_check = false
```

— Exact env var, config key and runnable commands provided. v0.22.1

03 **Cosign signature verification for uteke upgrade** NEW

80

`uteke upgrade` now verifies the cosign signature of each release itself, checking `checksums-sha256.txt.bundle`, the Fulcio certificate chain, the workflow identity (`release.yml` at the exact tag), and the Rekor signed timestamp — no manual signing step or key rotation required.

— Clear mechanism and named artifacts, but no runnable example given. v0.22.1

---

04 **Link legacy room metadata records to rooms** NEW

60

A consolidated room record stored with the old `room` metadata key can now be linked to its room using `uteke room link <id> <room>`.

— Exact command named but no example run or further context. v0.22.1

└──▷ **BREAKING ON UPGRADE**

! Builds before v0.22.1 cannot verify a keyless cosign-signed release and must be reinstalled once (`brew upgrade codecoradev/tap/uteke`, `cargo install uteke-cli`, or a manual download) before `uteke upgrade` will work again.

! The minisign signature of `checksums-sha256.txt`, `scripts/sign-release.sh`, `scripts/uteke-release.pub`, and the `minisign-verify` dependency are removed; `uteke upgrade` refuses any release older than v0.22.1.

WAS THIS USEFUL? 👍 👎

Pinecone

Feedback ⓘ announced · Oct 27, 2026 BLOG ↗ RANK

Pinecone is a managed vector database that stores and queries embeddings for AI applications.

> Pinecone launched pg_pinecone, an open-source Postgres extension that brings filtered vector search, switchable search engines, in-database embedding generation, BM25 full-text search, and reranking directly into Postgres.

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

01 Filtered vector indexing in Postgres NEW 85

pinecone.create_index('docs', 'embedding', filter_column => 'tenant_id') builds a vector search index with pre-filtering support so narrow per-tenant queries still return a full result set.

Index an existing Postgres table with a per-tenant filter column so narrow queries still return a full set of results.

```
$ CREATE EXTENSION pg_pinecone;
SELECT pinecone.create_index('docs', 'embedding', filter_column
=> 'tenant_id');
SELECT id, body FROM docs WHERE tenant_id = $1 ORDER BY embedding
<=> $2 LIMIT 10;
```

— Names exact function, args, and runnable SQL example launch-20261027-43b85fd7

02 Switchable search engines via set_engine() NEW 85

pinecone.set_engine() switches a table between three search engines — `native`, `pgvector`, and `remote` — without changing the underlying SQL queries, letting you move from a local index to Pinecone's managed service as data grows.

Move search to Pinecone's managed service when the local index outgrows available memory, without changing application SQL.

```
$ SELECT pinecone.set_engine('docs', 'remote');
```

— Names all three engines and gives a runnable command launch-20261027-43b85fd7

03 Result reranking via rerank() NEW 75

pinecone.rerank() reorders vector or merged search results by relevance using Pinecone's reranking models, useful for RAG pipelines reordering top candidates before passing them to an LLM.

Rerank the top-20 vector results so the most relevant records appear first before passing them to an LLM.

```
$ SELECT * FROM pinecone.rerank(
  'which database stores vectors?',
  ARRAY(
    SELECT content FROM docs ORDER BY embedding <=> $1 LIMIT 20
  )
);
```

— Named function paired with a runnable reranking query launch-20261027-43b85fd7

04 **In-database embedding generation** NEW 65

pinecone.embed(content, input_type => 'passage') generates embeddings from text columns directly in SQL by calling Pinecone's inference service, with no need to host a model.

— Function named but no standalone runnable example given launch-20261027-43b85fd7

05 **BM25 full-text search alongside vectors** NEW 60

pinecone.track_docs('docs', text_columns => '["content"]') and pinecone.search_docs() enable BM25 full-text search on the same table as vector search.

— Functions named but no worked example provided launch-20261027-43b85fd7

THINNER COVERAGE BELOW

06 **Installation via GitHub release or Docker Hub** NEW 55

pg_pinecone can be installed from a pre-built release on GitHub or via the `pineconeteam/pg_pinecone` Docker Hub image; the `native` engine requires no Pinecone account or API key.

— Install sources named but no exact command given launch-20261027-43b85fd7

WAS THIS USEFUL? 👍 👎

Weaviate

Feedback ⓘ announced · Oct 14, 2026 BLOG ↗ RANK

Weaviate is an open-source vector database that stores both objects and vectors, allowing for the combination of vector search with structured filtering with the fault tolerance and scalability of a cloud-native database.

> Weaviate 1.40 ships a preview Search REST API with six JSON endpoints and reference-following support, GA 4-bit Rotational Quantization, multi-vector HFresh indexing via MUVERA, a new PathSeer HNSW filter strategy, and the ability to drop a named vector index without recreating the collection.

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

01 Search REST API in preview with reference following NEW

95

Six Search REST API endpoints are now available in preview (on by default from v1.39.7): `POST /v1/search/{collection}/near-text`, `/bm25`, `/hybrid`, `/near-object`, `/near-vector`, and `POST /v1/aggregate/{collection}`, accepting plain JSON over HTTP/1.1 with no GraphQL or generated client required. Requests can include a `returnReferences` field to follow named references via `linkOn`, `targetCollection`, `returnMetadata`, and nested `returnReferences`, replacing the now-rejected dotted `hasAuthor.name` form previously used in `returnProperties`.

Run a near-vector search and follow a reference to each result's author using the new Search REST API, with no GraphQL required.

```
$ curl -s -X POST http://localhost:8080/v1/search/Book/near-
vector \
  -H 'Content-Type: application/json' \
  -d '{"vector":[0.1,0.9,0.2],"limit":1,"returnProperties":
["title"],"returnReferences":
[{"linkOn":"writtenBy","returnProperties":
["name"]}],"returnMetadata":["distance"]}'
```

— Six endpoints and fields named with a runnable curl example launch-20261014-72a86821

02 Named vector index deletion IMPROVED

80

The `DELETE /v1/schema/{collection}/vectors/{name}/index` endpoint, or the Python client's delete_vector_index() method, now lets you drop a named vector index to reclaim disk and memory without re-creating the collection — the capability is now on by default.

Remove an unused named vector to reclaim disk and index memory after switching to a better embedding model.

```
$ curl -X DELETE
http://localhost:8080/v1/schema/Article/vectors/body_vector/index
```

— Endpoint and method named with a runnable curl command launch-20261014-72a86821

03 **PathSeer HNSW filter strategy** NEW 80

Weaviate adds `pathseer` as a third `filterStrategy` option for HNSW indexes, selectable via `VectorFilterStrategy.PATHSEER` in the Python client or `'filterStrategy': 'pathseer'` in REST; switching requires no reindexing and no new parameters to tune.

Switch an existing HNSW collection to the PathSeer filter strategy for faster filtered vector search without reindexing.

python

```python
articles = client.collections.use('Article')
articles.config.update(
    vector_config=Reconfigure.Vectors.update(
        name='default',
        vector_index_config=Reconfigure.VectorIndex.hnsw(
            filter_strategy=VectorFilterStrategy.PATHSEER,
        ),
    )
)
```

— Named strategy value with a runnable reconfiguration example launch-20261014-72a86821

04 **4-bit Rotational Quantization reaches GA** NEW 75

Setting `DEFAULT_QUANTIZATION=rq-4` makes 4-bit Rotational Quantization the default for every new HNSW index; 4-bit RQ is now generally available and compresses a 1536-dimension vector from 6144 bytes to 784 bytes (7.84x).

— Config key and compression numbers given, no runnable example launch-20261014-72a86821

05 **Multi-vector embeddings in HFresh via MUVERA** NEW 70

HFresh, the disk-based vector index, now accepts multi-vector embeddings by setting `multivector.muvera.enabled` and configuring Configure.MultiVectors.self_provided() with Configure.VectorIndex.MultiVector.Encoding.muvera(), enabling ColBERT and ColPali embeddings on a disk-based index.

— Config fields and functions named, no usage example given launch-20261014-72a86821

THINNER COVERAGE BELOW

06 **Vector configuration defaults changed for DEFAULT_VECTORIZER_MODULE servers** 35

BREAKING

If the server sets `DEFAULT_VECTORIZER_MODULE`, vector configuration defaults have changed in this release — review the Vector configuration defaults section of the release notes before upgrading.

— Only a general pointer to review notes, no specifics launch-20261014-72a86821

└──▷ **BREAKING ON UPGRADE**

! If the server sets `DEFAULT_VECTORIZER_MODULE`, vector configuration defaults have changed — review the Vector configuration defaults section of the release notes before upgrading.

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

Braintrust

Feedback ⓘ changelog · Oct 31, 2026 CHANGELOG ↗

RANK

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

Braintrust adds async SQL queries (`bt sql --async`), a Terraform provider, `bt mcp` CLI tool access, TypeScript application logs, Loop GitHub integration, and MCP human-review queueing.

Run async SQL queries that execute in the background and return results in pages using `bt sql --async`, so large exports and long-running reads are no longer limited to a single synchronous request.

NEW INTEGRATION List and run Braintrust MCP server tools without an MCP client using `bt mcp` (public preview), passing tool arguments as generated flags or as JSON input.

Trace Muse Code sessions — including turns, model calls, tool calls, and subagents — by running `bt trace enable muse`; also supports `bt trace run muse` for a single invocation and `bt trace import muse` for past or active sessions (requires Muse Code 1.3.0 or later).

Import compressed Codex transcripts with `bt trace import codex`, which reads `.jsonl.zst` session files that Codex has compressed.

Fetch Braintrust documentation pages by URL using `bt docs fetch`, which now recognizes links on `docs.braintrust.dev` in addition to `www.braintrust.dev/docs`.

+16 more

NEW INTEGRATION MCP server tracing — TypeScript SDK via `wrapMCPServer()`

NEW INTEGRATION Cloudflare Workers AI tracing via `wrapWorkersAI()`

└──▷ BREAKING ON UPGRADE

- ! MCP content capture configuration in the Python SDK moves from MCPIntegration.setup(capture_content=False) and per-object wrappers to MCPIntegration.setup(); the old per-object API no longer applies.
- ! `AskUserQuestion` spans in Claude Code are now task spans instead of tool spans — queries that select them as tool spans will return no results and must be updated.
- ! Resuming a Claude Code session that has already ended now starts a new trace with the same `session_id` instead of continuing the original trace.
- ! The LiveKit Agents integration in the TypeScript SDK no longer records audio automatically; set `BRAINTRUST_CAPTURE_USER_AUDIO_ATTACHMENTS` and `BRAINTRUST_CAPTURE_AGENT_AUDIO_ATTACHMENTS` explicitly to capture audio.

WAS THIS USEFUL? 👍 👎

LangChain LangSmith

Feedback ⓘ updated · Oct 31, 2026

NOTES ↗

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

LangSmith adds MCP tools for querying runs, managing prompts, and accessing sandbox files, plus new deployment log fields, a dataset bulk-replace API, and LSU-based Engine pricing.

The LangSmith MCP adds a `query_runs` tool that accepts the same parameters as the SDK's client.runs.query(), including typed selects for costs, token breakdowns, and full inputs and outputs, with cursor pagination; it returns token counts and USD costs by default.

New MCP tools let agents push prompt versions, browse commits, manage commit tags, update or delete prompts, and read and commit Context Hub agent and skill files.

The LangSmith MCP can read, write, and search sandbox files, stream long-running sandbox commands, share sandbox ports behind LangSmith login, and report sandbox usage, organization usage, audit logs, usage limits, trace retention settings, and organization members.

Sandbox proxy rules now accept `match_headers` entries such as `'authorization: Bearer account-b'`, so several rules on one host can each swap a different placeholder for its own credential and serve multiple accounts.

Agents connected to LangSmith's native MCP server can find deployments and revisions and read build or runtime logs, filtered by revision, time, search text, and severity.

+7 more

NEW INTEGRATION Vertex AI Model-as-a-Service open model cost resolution

NEW SUPPORTED PLATFORM GPT-6.1-Sol in Playground and model configurations

NEW SUPPORTED PLATFORM Claude Sonnet 5.5 in Playground and model configurations

NEW SUPPORTED PLATFORM Gemini 3.7 Flash and Gemini 3.8 Flash in model selectors

UPDATED SUPPORTED PLATFORM OpenAI all-models pricing page cost resolution expanded

WAS THIS USEFUL? 👍 👎

overmind

Feedback ⓘ v0.1.23 NOTES ↗ RANK

The platform for continuously improving AI agents.

> overmind's v0.1.23 release adds MCP-based credit balance reporting, more memory-efficient FP8 KV cache inference, deeper trace observability, and capability-free model handoffs.

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

01 **Credit balance and usage reporting over MCP** NEW 55

Account credit balance and usage can now be reported over MCP, making balance data queryable directly from coding agents like Cursor and Claude Code.

— Names MCP surface and client agents but no exact method/field v0.1.23

02 **FP8 KV cache inference with per-model sequence cap** IMPROVED 50

Chat serving now uses an FP8 KV cache along with a per-model sequence cap, intended for more memory-efficient inference.

— Names mechanism but no config/flag to invoke it v0.1.23

03 **Capability-free model handoffs** NEW 40

Model handoffs can now occur without a capability, allowing agent control-flow transitions that are not tied to a defined capability node.

— Explains mechanism but no concrete example or config v0.1.23

04 **Inspectable trace scores, payloads, and errors** NEW 35

Trace scores, payloads, and errors are now inspectable, giving deeper observability into production agent runs.

— States capability but no UI path or API named v0.1.23

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

julius

Feedback ⓘ v1.5.5 NOTES ↗ RANK

Simple LLM service identification - translate IP:Port to Ollama, vLLM, LiteLLM, or 60+ other AI services in seconds

> julius's probe command now flags authentication-required LLM endpoints, helping practitioners distinguish protected services from open ones during fingerprinting scans.

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

01 **Auth-required signal in probe fingerprinting** NEW 45

The `probe` command now surfaces an auth-required signal in its fingerprint output, letting practitioners distinguish protected LLM endpoints from open ones during service discovery.

– Names the command and signal but no output format or example. v1.5.5

WAS THIS USEFUL? 👍 👎

cplt

Feedback ⓘ 2026.10.30-110556-eab7fb6 NOTES ↗ RANK

cplt is a kernel-level sandbox for AI coding agents, running Copilot CLI, Claude Code, OpenCode, Gemini CLI, and others with git and gh guards.

> cplt added a direct `config unset` command for clearing configuration keys.

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

01 **config unset command** NEW 60

A new `config unset` command clears a configuration key, mirroring the existing `config set --unset` behavior.

– Names exact command but no further mechanism detail 2026.10.30-110556-eab7fb6

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Open-Weight Models

ConvAI Innovations Laya

Feedback ⓘ v0.4.19 NOTES ↗

RANK

Non-autoregressive System 1 decision engine. Typed choice, score and yes/no decisions over any text in a single forward pass, in 100+ languages, with a router that picks the right checkpoint per request.

> Laya shipped an ONNX Runtime backend for its C++ inference layer and added routing diagnostics surfaced both over HTTP and as an opt-in MCP feature.

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

01 **ONNX Runtime backend for inference** NEW 60

Laya's C++ layer adds an ONNX Runtime backend (`laya_onnx`) and an `onnx` lane for running inference through ONNX models.

– Names backend and lane but no usage command or config shown v0.4.18

THINNER COVERAGE BELOW

02 **Routing diagnostics over HTTP and MCP** NEW 47

Laya exposes server-side routing diagnostics over HTTP via a new diagnostic endpoint added to the serve component, and adds an opt-in routing diagnostics feature for MCP (introduced in PR #1409).

– Describes two diagnostic surfaces but no endpoint path or flag given v0.4.19 v0.4.18

WAS THIS USEFUL? 👍 👎

FrontiersMind Lumma-fev-0.1B-v2

Feedback ⓘ

released · Oct 31, 2026 MODEL ↗

Lumma-fev-0.1B-v2 from FrontiersMind was released on 30 Oct 2026. Open weights on Hugging Face.

WAS THIS USEFUL? 👍 👎

H2O.ai

Feedback ⓘ announced · Oct 29, 2026 PRESS ↗ RANK

H2O LLM Studio - a framework and no-code GUI for fine-tuning LLMs. Documentation: https://docs.h2o.ai/h2o-llmstudio/

> H2O.ai introduced H2O-3 Secure, a new commercially licensed, enterprise-grade edition of H2O-3 aimed at regulated industries, bundling governance, compliance, and production deployment capabilities.

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

01 **H2on-3 Secure enterprise edition launch** NEW 60

H2O.ai launched H2O-3 Secure, a commercially licensed edition of H2O-3 that adds role-based access control, tamper-evident audit logs, and end-to-end model lineage for governance and compliance programs. It includes H2O AutoDoc for automated model documentation and built-in drift monitoring to track production models, and can be deployed standalone or as part of H2O AI Cloud, targeting SOC 2, ISO 27001, and ISO 42001 compliance frameworks. The edition supports production deployment on Hadoop, Spark, and Kubernetes with enterprise single sign-on and access to Enterprise algorithms including the Managed XGBoost Service, plus commercial support and structured security maintenance for MOJO scoring artifacts used in fraud, credit, and pricing systems.

— Names many capabilities but no commands, flags, or endpoints to act on.

launch-20261029-b2cbde29

WAS THIS USEFUL? 👍 👎

DeepSeek Models

Feedback ⓘ Date: 2026-10-17 NOTES ↗ RANK

Deepseek models is the DeepSeek API uses an API format compatible with OpenAI/Anthropic. By modifying the configuration, you can use the OpenAI/Anthropic SDK or softwares compatible with the OpenAI/Anthropic API to access the DeepSeek API.

> DeepSeek shipped a new in-house Search API and upgraded the deepseek-flash model endpoint to a faster, cheaper version.

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

01 **deepseek-flash upgraded to V4.1-Flash-1017** IMPROVED

60

The `deepseek-flash` model endpoint now points to DeepSeek-V4.1-Flash-1017, with reduced task runtime and output length at unchanged pricing.

— Names exact endpoint and model version but no benchmark numbers Date: 2026-10-17

THINNER COVERAGE BELOW

02 **In-house Search API** NEW

55

A new Search API gives access to DeepSeek's general-purpose search engine, which indexes hundreds of billions of web pages, supports multilingual search, and filters SEO spam, GEO manipulation, and low-quality AI-generated content.

— Describes scope and filtering but no endpoint path or request format Date: 2026-10-17

WAS THIS USEFUL? 👍 👎

◆ Frontier Models

Google Gemini API

Feedback ⓘ October 30, 2026 NOTES ↗

RANK

Gemini API provides Gemini models for generating text, images, and other multimodal content in applications.

> Gemini API moved its Computer Use tool to general availability, now accessible through both the Interactions and GenerateContent APIs.

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

01 **Computer Use tool reaches general availability** IMPROVED

35

The Computer Use tool is now generally available across the Interactions and `GenerateContent` APIs.

— Names the APIs but no usage details or mechanism given. October 30, 2026

WAS THIS USEFUL? 👍 👎

Mistral AI Mistral Large 4

Feedback ⓘ

price cut · Oct 30, 2026 MODEL ↗

Mistral Large 4 is a multimodal model from Mistral AI.

Price cut on OpenRouter (mistralai/mistral-large-4-0): $1.36 → $0.70 in, $4.18 → $2.10 out per million tokens (-50%), first seen 30 Oct 2026.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · October 31, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue