NOW AVAILABLE The 512GB Mac Studio is on sale now, shipping February to March. See every configuration.

THE AI TOOLCHAIN NO. 081

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED NOVEMBER 2, 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 159 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 50 of 52 tools

▾ // THE BRIEF

HIDE SAME ISSUE, SAME PROMPT, TWO WRITERS: GPT-5.5 | Opus 5

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

vim ? help

Three unrelated things stand out today: SkillSpector scans more MCP and Python paths for malware and credential theft; OpenViking can run its gateway across replicas with Redis-backed sessions; memwhale can abstain when retrieval has no evidence instead of feeding junk context downstream.

GOVERN

Catch agent-tool supply-chain risks before they become runtime permissions

SkillSpector broadened detection across MCP configs, Pipenv dependencies, and urllib taint paths, with configurable scan limits via environment variables. That gives teams a preflight check for malicious tools and credential-exfiltration paths before an agent is allowed to load them.

SkillSpector

DEPLOY

Run agent gateways as shared infrastructure instead of a single sticky process

OpenViking added a multi-replica Redis-backed Gateway with configurable session storage, plus self-service API key rotation. Teams can scale and restart the gateway without tying sessions to one process, and users can rotate access without waiting on an operator.

OpenViking

OPERATE

Stop no-match retrieval from turning into confident wrong context

memwhale now returns an explicit no-evidence signal when search has no real matches. Instead of handing unrelated rows to a model and hoping prompting suppresses them, downstream code can branch on abstention and ask for clarification, fall back, or decline.

memwhale

EVALUATE

Exercise real automation contracts against a sandbox every night

ForgeDock added a live contract selftest harness that runs actual gh/git/pool operations against a sandbox repo and now schedules it nightly via the schedd timer. That catches integration drift in the tools an agent depends on before the next production run hits the broken contract.

forgedock

Does **GPT-5.5** read better? 👍 👎

BUILD

◆ AI Coding Agents

yetone magpie

Feedback ⓘ v0.1.2703 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 this window expanded far beyond basic model-swapping: it added dozens of new client and provider presets, started archiving its internal System One classifier calls, shipped a Pi codemode extension exposing its models as providers, fixed Codex tool-name rejections with an automatic translation layer, and improved routing and key management with free-model tagging, decision-model key binding, and automatic image trimming for oversized requests.

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

01 **Decision models selectable in key editor** NEW 85

The key editor now lists decision models (Jev, Clef) alongside a provider's chat models, marked via the `decides` field in `keyModelJSON`, so a gateway key can be held to a decision model directly without needing `All models`. The `/api/caller-keys/models` endpoint now returns decision models in its model list, each carrying `"decides": true` to distinguish them from chat models.

Check which models are available to assign to a key, and confirm decision models (Jev, Clef) now appear in the list.

```
$ curl -s http://localhost:<port>/api/caller-keys/models | jq
'.Models[] | select(.decides == true)'
```

— Named endpoint, field, and a runnable curl example v0.1.2681

02 **Pi codemode extension exposing magpie models** NEW 80

magpie now writes `extensions/magpie.ts` into Pi's agent folder, registering its decision models as the `magpie-systemone` provider (via `/v1/systemone`) and its image models as the `magpie-images` provider (via `/v1/images/generations`), so Pi's codemode can call models.classify() and models.generateImages() using magpie-configured models.

— Named file and endpoints give a clear integration point v0.1.2699

03 **Automatic tool-name translation for Codex backend** NEW 75

Adds `CodexToolName` in `internal/provider/codex_names.go` to translate any tool name the ChatGPT backend rejects (outside `^[a-zA-Z0-9_.-]+$`) by replacing disallowed characters with `_`, truncating to 55 characters, and appending a SHA-256 hash suffix for stable, unique names. Adds `CodexToolNamesBack` to map translated names back to the agent's original declared names, and `codexNamesBack` / `codexRenamedIn` in `internal/gateway/codex_names.go` apply this translation to both gateway-decoded replies and raw Responses-API replies relayed through `nsTidy`.

— Mechanism fully named but happens automatically, not user-invoked `v0.1.2694`

---

04 **Request archive covers System One classifier calls** IMPROVED

60

The request archive now also records System One (`/v1/systemone`) calls that magpie makes internally to a Jev classifier, whereas previously only client-sent chat requests were archived. The Routing response now indicates when the request archive is enabled.

— Clear before/after but no user-facing action given `v0.1.2703`

THINNER COVERAGE BELOW

05 **New AI client and provider presets** NEW INTEGRATION NEW

58

magpie added support for dozens of new AI coding clients as proxy targets, including Cursor Private Inference, Kiro Crew, T3 Code, Snow CLI, `AstrBot`, `QwenPaw`, and `OpenClaw`. It also added `Huawei Cloud MaaS`, Volcengine Ark, Nous Portal, and `LongCat` as built-in provider presets, and now supports any OpenAI- or Anthropic-compatible URL as a provider preset target for routing to arbitrary self-hosted or third-party endpoints.

— Names many integrations but no setup steps shown `product docs`

---

06 **Free-model indicator in routing groups** NEW

55

Group member metadata now exposes a `free` field (in `memberJSON`) that is `true` when a member's provider is marked Free for me or the vendor serves it free, letting Free first and Cheapest first routing identify zero-cost models. The routing page member label now appends a free tag alongside fast for members whose `memberJSON.Free` is true.

— Names the field and UI tag; points to routing page `v0.1.2687`

---

07 **Automatic image trimming for oversized requests** NEW

45

Requests that exceed a provider's byte limit (Anthropic, DeepSeek, Command Code, and others) are now automatically retried with the oldest inline images stripped out and

replaced by short text placeholders, leaving the latest user message's images and all text untouched.

— Explains behavior and providers but is fully automatic v0.1.2679

v0.1.2680 CODE ↗ STABLE

magpie adds LongCat (Meituan) as a vendor preset supporting chat completions, Responses, and Anthropic messages.

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.2680
```

git

```
$ git clone --branch v0.1.2680 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.2680
```

published under the project's GitHub account · checked 2 Nov Wrong command?

WAS THIS USEFUL? 👍 👎

Letta Code

Feedback ⓘ v0.34.38 NOTES ↗ RANK

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

> Letta Code improved visibility into subagent task execution, adding real-time progress streaming into task output files and a complete JSONL event log for every subagent task, including remote workers.

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

01 **Subagent task streaming and JSONL event logs** IMPROVED

44

Subagent task output files now receive streamed progress from Letta subagents and external workers in real time as work runs, and every subagent task produces a full JSONL event log covering both remote and local workers.

— Describes mechanism but no file path, flag, or command shown v0.34.38 v0.34.37

WAS THIS USEFUL? 👍 👎

clay

Feedback ⓘ v5.5.0 NOTES ↗ RANK

Clay is a self-hosted workspace where people and coding agents work together across projects and sessions.

> Clay's v5.5.0 release consolidates provider and worktree management into a unified settings area, adds personal agent behavior controls, and introduces dynamic pane management in split-view.

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

01 Unified settings for providers and worktree controls IMPROVED

55

Provider setup and Worker/worktree controls let users manage AI providers and worktree sessions from within the UI; this was combined with AI preferences into a single settings area, replacing the previously separate views.

– Names settings areas merged but no exact config keys or paths v5.5.0

02 Pane insertion and removal in split-view NEW

50

Split-view now supports pane insertion and removal controls, enabling dynamic workspace layout management.

– Describes the capability and UI area, no exact controls named v5.5.0

03 Personal agent behavior controls NEW

35

New personal agent behavior controls under settings let users configure how the agent acts on their behalf.

– States purpose but no mechanism or specific options listed v5.5.0

04 Workspace navigation and appearance refinements IMPROVED

5

General refinements to workspace navigation and appearance across the UI.

– Vague catch-all with no specifics v5.5.0

WAS THIS USEFUL? 👍 👎

coffee-cli

Feedback ⓘ v3.8.6 NOTES ↗ RANK

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

> coffee-cli's two releases add isolated worktrees and richer multi-agent coordination for Claude Code, Codex, OpenCode, and DeepSeek Harness, plus custom notification sounds, resume indicators, and usage-display improvements.

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

01 **Custom notification sounds in Settings** NEW 90

Set custom MP3 or WAV notification sounds (up to 20 MB, plays up to 10 seconds) for task completion and permission requests separately in `Settings > Sound`, with automatic fallback to the default sound if a file can't play.

– exact UI path, file formats, size and duration limits given v3.8.4

02 **Worktrees for agent conversations** NEW 65

Enable worktrees for new Claude Code, Codex, OpenCode, or DeepSeek Harness conversations to work in an isolated repository copy on its own branch, with one-click merge back and no automatic cleanup of unmerged work.

– mechanism described but no exact command or menu path v3.8.4

03 **Multi-agent coordination and worker control** IMPROVED 60

Multi-agent work now supports choosing the coordinator's role, running workers in their own worktrees, steering workers while they run, automatic retry after transient service errors, and coordinator awareness of tasks interrupted by a restart.

– lists capabilities but gives no commands or UI path v3.8.4

THINNER COVERAGE BELOW

04 **Return indicator when resuming agent conversations** IMPROVED 55

Resuming a Claude Code, Codex, OpenCode, or DeepSeek Harness conversation now shows each tool's own working animation with a Returning to the conversation message instead of a static welcome screen.

– names affected tools but no config or toggle described v3.8.6

05 **Usage display improvements for keys and vendors** IMPROVED 45

Unlimited `new-api` keys now display how much credit has been spent, where previously no usage figure was shown; usage display also now recognizes more relay services and model vendors.

— names the key type but no numbers or vendor list v3.8.6 v3.8.4

06 **Latest-first rendering for long Claude Code history** IMPROVED

35

Opening a long Claude Code conversation from history now renders the latest messages first instead of waiting for the full history to load.

— describes behaviour change without metrics or scope v3.8.6

07 **Auto-growing composer input box** IMPROVED

15

The composer input box now grows taller automatically as your draft gets longer.

— single-line description with no further mechanism v3.8.4

WAS THIS USEFUL? 👍 👎

pstack-claude

Feedback ⓘ v0.9.116 NOTES ↗ RANK

pstack-claude is a port of Poteto's pstack agent framework to Claude Code, Codex, Copilot, Pi, and other AI coding assistants.

> pstack-claude brought Windows support to the `statechart` command and added configurable model selection for its thermo-nuclear reviewer role.

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

01 **Windows support for statechart command** NEW 80

The `statechart` command now runs on Windows: it kills the TLC worker's process tree with `taskkill` and delegates cleanup to a new `janitor.ts` module instead of `/bin/sh`, which Windows Node cannot find. The worker's `TEMP` and `TMP` environment variables now point at the command's temp folder, matching `TMPDIR`, so TLC's working files land where cleanup after Ctrl-C can reach them. A new `windows-latest` CI job charts the `MediaPlayer` corpus spec and diffs its `report.md` and `chart.mmd` output against the Linux results.

— Names exact mechanism, modules and CI job but no user-facing command flag v0.9.116

02 **Configurable thermo-nuclear reviewer model** BREAKING 77

A new `thermos reviewers` line in `pstack-models.md` lets users control which model both thermo-nuclear reviewers use, defaulting to the `default` tier (`opus`). `/setup-pstack` now lists this role and stamps a Models section into the skill. Breaking: a `pstack-models.md` sheet written before this release lacks the `thermos reviewers` line, so on GitHub Copilot the session hook reports `sheet invalid`, drops the sheet's role lines, and the next skill needing a role model must run `setup-pstack` first.

— Names config key, default tier and exact migration/error behavior v0.9.115

└──▷ **BREAKING ON UPGRADE**

! On GitHub Copilot, a `pstack-models.md` sheet written before this release has no `thermos reviewers` line; the session hook reports `sheet invalid`, drops the sheet's role lines, and the next skill that needs a role model runs `setup-pstack` first.

WAS THIS USEFUL? 👍 👎

Token Optimizer

Feedback ⓘ v5.13.63 NOTES ↗ RANK

Token Optimizer is a coding-agent plugin that trims wasted context, checkpoints work across sessions and compactions, and reports token spend on a live dashboard.

> Token Optimizer's v5.13.63 release brings Windows support to more coding agents, expands the Hermes dashboard with richer session details, adds new spend-tracking metrics, and fixes a bug where post-compaction file reads were truncated.

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

01 **Windows support for Antigravity, Cursor and Grok** BREAKING 67

Windows installs now support Antigravity, Cursor and Grok. Antigravity users on Windows must run `antigravity-install` once after updating to complete the install.

— Names exact tools and required command, but no detail on what Windows support entails. v5.13.63

THINNER COVERAGE BELOW

02 **Full file content on first read after clear/compaction** IMPROVED 55

The first file read after `/clear` or a compaction event now returns the full file content, rather than a truncated or partial version.

— Names exact trigger conditions, behavior change is clear and actionable. v5.13.63

03 **Hermes dashboard shows skills, MCP servers and memory** IMPROVED 45

The Hermes dashboard now shows skills, MCP servers, `SOUL.md` and memory, with Hermes labels applied to these sections.

— Names concrete dashboard fields but no path or mechanism given. v5.13.63

04 **'Money on the table' routing estimate** NEW 45

A new 'money on the table' routing estimate assumes about 20% of Opus work could move to a cheaper model, and is shown separately from the savings total on the dashboard.

— States the assumption and model named, but no mechanism for the estimate. v5.13.63

05 **Cumulative total-since-install savings figure** NEW 35

A cumulative `total since install` figure is now tracked on the dashboard, using the same four components as the existing monthly savings number.

— References unnamed 'four components', limiting specificity. v5.13.63

└──▷ BREAKING ON UPGRADE

! Antigravity users on Windows must run `antigravity-install` once after updating to complete the install.

WAS THIS USEFUL? 👍 👎

codealta

Feedback ⓘ 3.3.0 NOTES ↗ RANK

Agentic Development Environment

> CodeAlta 3.3.0 lets agents drive real browsers, run Claude Code sessions in the cloud, profile .NET applications with a built-in skill, and search or change settings via a new `alta settings` command.

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

01 **.NET profiler skill (ultra-profiler)** NEW 90

Profile slow .NET, mixed, or native applications by asking an agent to use the built-in `ultra-profiler` skill, which runs Ultra via `dotnet dnx` and captures CPU, GC, and JIT events as a background job.

Ask an agent to profile a slow .NET service using the built-in profiler skill.

```
$ altacli run "Profile the startup of MyService.dll with ultra-profiler and show me the hot functions"
```

— Named skill, runtime command and metrics, with runnable example 3.3.0

02 **Browser automation tools for agents** NEW 75

Agents can drive Chrome, Edge, Firefox, or Safari 27 and later via browser tools — open pages, click, fill forms, take screenshots, and read the console and network — with no Node.js or driver to install. Browser behavior is configured in Settings > Extensions > Browser, including persistent profiles or use of your own browser, with sessions restricted to the tabs they opened.

— Names browsers, UI path and restrictions; no code sample given 3.3.0

03 **Cloud Claude Code sessions** NEW 65

Run Claude Code in the cloud by choosing Cloud in the Workspace chip, so sessions continue on Anthropic's machines while your computer sleeps and return with their branch. A cloud session started on claude.ai can be brought back to the desktop with Bring a cloud session here, returning the full conversation and branch to a new worktree or the project folder.

— Explains mechanism and UI entry points, no runnable command 3.3.0

04 **Settings search and `alta settings` command** NEW 65

Search settings from `Ctrl+P` or the Settings search field, and let agents open or change settings with the `alta settings` command, with an Undo toast shown in the window. Plugins can now declare their own settings, which appear in Settings search alongside built-in settings.

.

— Named shortcut and command, concrete UX detail, no sample invocation `3.3.0`

WAS THIS USEFUL? 👍 👎

Superset

Feedback ⓘ desktop-v2.0.3 NOTES ↗ RANK

Superset is a desktop workspace for running CLI coding agents like Claude Code with terminals, diff viewer, and browser previews.

> Superset's 2.0.3 release adds per-turn undo for agent work, richer chat rendering with Mermaid and HTML, GitHub-synced organization plugins, and new automation grouping and workspace-status controls.

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

01 **Automation grouping and bulk-edit** NEW 83

Automations can now be grouped, dragged, and bulk-edited in the desktop UI, backed by `automations.group_name` in the database, a `--group` flag on the CLI, and `setGroup` plus an MCP group in the API.

— Names DB field, CLI flag, and API method but no example desktop-v2.0.3

02 **CLI control of sidebar panes** NEW 70

Agents can drive the right sidebar from the CLI using `superset panes`.

Have an agent open a specific right-sidebar pane programmatically during a workflow.

```
$ superset panes <pane-name>
```

— Exact runnable command provided desktop-v2.0.3

THINNER COVERAGE BELOW

03 **Account-level chat settings storage** IMPROVED 57

Adds account-level Chat settings stored in a `user_settings` table, replacing PostHog for that preference.

— Names table and prior mechanism replaced desktop-v2.0.3

04 **GitHub-synced organization plugins** NEW 55

Organization plugins now sync from `GitHub` and are offered per role; new `Peec AI` and `Sazabi` plugins are added.

— Names plugins and sync source but no setup steps desktop-v2.0.3

05 **Per-turn undo for agent changes** NEW 50

A new per-turn changes card tracks turn snapshots and lets you roll back individual agent turns with Undo.

— Explains mechanism but no UI path or command desktop-v2.0.3

06 **Cloud workspace status and link sharing** NEW 45

The sidebar now shows cloud workspace connection status and remote host offline state, and lets you copy a workspace link, which can also be printed with the CLI.

— Mentions CLI printing but no exact command desktop-v2.0.3

07 **Mermaid and HTML rendering in chat** NEW 40

Chat replies now draw Mermaid diagrams, preview HTML, and let you copy tables.

— Clear capability but no config or command detail desktop-v2.0.3

08 **Scheduled and event-based Dev routines** NEW 30

Adds Dev routines so a Dev agent wakes on a schedule or an event.

— No trigger syntax or config surface given desktop-v2.0.3

09 **Link menu on terminal URL click** IMPROVED 20

Opens a link menu on plain click of a terminal URL.

— Bare description of a small UI change desktop-v2.0.3

10 **Read-only shell command labeling** IMPROVED 20

Shell commands that only read, search, or list are now labeled as such in the UI.

— Bare description with no mechanism or location desktop-v2.0.3

WAS THIS USEFUL? 👍 👎

octomind

Feedback ⓘ 0.68.0 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's supervisor now grounds its alarm advice in specific, traceable evidence rather than generic alerts.

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

01 **Evidence-bound alarm advice in supervisor** IMPROVED 30

The supervisor component now issues alarm advice that is bound to specific evidence, grounding alerts in traceable source material.

— Describes behaviour change but no mechanism, config, or example given. 0.68.0

WAS THIS USEFUL? 👍 👎

forgedock

Feedback ⓘ v1.42.0 NOTES ↗ RANK

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

> ForgeDock introduces a live contract selftest harness that exercises real gh/git/pool operations against a sandbox repo, now also run automatically every night via the schedd timer.

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

01 Live contract selftest harness with nightly scheduling NEW

77

Setting `selftest.sandbox_repo` in config (e.g. in `forge.yaml`) points the live contract harness at a throwaway sandbox repo so it can verify real gh/git/pool operations; this harness blocks release review until it passes. forgedock now also runs these live selftests automatically every night through the `schedd` timer, keeping the harness exercised between releases.

Point the selftest harness at a throwaway sandbox repo so the release gate can verify real gh/git/pool operations before promotion.

yaml

```yaml
selftest:
  sandbox_repo: RapierCraftStudios/forgedock-selftest-sandbox
```

— Names config key, timer and gating behaviour with example v1.42.0

WAS THIS USEFUL? 👍 👎

zot

Feedback ⓘ v0.4.60 NOTES ↗ RANK

zot manages coding agent workflows from the command line.

> zot's v0.4.60 release makes a small TUI readability tweak, dimming redundant working-directory prefixes in Bash command output.

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

01 Dimmed working-directory prefixes in Bash output IMPROVED

30

zot's TUI now dims redundant working-directory prefixes shown in Bash command output, reducing visual noise when reviewing command results.

— Describes the visual change but no config or toggle named. v0.4.60

WAS THIS USEFUL? 👍 👎

ogcode

Feedback ⓘ v0.57.0 NOTES ↗ RANK

An open-source, local-first AI teammate and agent operating environment for turning intent into completed work—with tools, memory, permissions, and human oversight.

> ogcode ships a per-project tool-call audit log you can query from chat, alongside a rebuilt Agents board view and a renamed, expanded built-in knowledge skill.

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

01 Tool-call audit log in `.ogcode/tool_calls.csv` NEW

88

Every finished tool call — built-in and MCP tools alike — appends one row to `.ogcode/tool_calls.csv` with columns `at,session,agent,tool,target,status,duration_ms,model,provider,error`. `status` values are `completed`, `error`, `denied`, `cancelled`, `disallowed`, `unknown`, or `panic`, matching the transcript's per-tool states so rows join with the session database. Set `OGCODE_TOOL_AUDIT=off` to disable logging, e.g. in a read-only project directory.

Disable the tool-call audit log when the project directory is read-only or you want to suppress the warning.

```
$ OGCODE_TOOL_AUDIT=off ogcode
```

— Names file path, every column, status values, and disable flag. v0.57.0

02 Query the audit log with `query_sheet` NEW

77

The `query_sheet` tool can load `.ogcode/tool_calls.csv` as `Sheet1` from inside the chat, letting you run SQL like `SELECT tool, status, COUNT(*) FROM Sheet1 GROUP BY tool, status` without leaving the session.

Tally every tool call by tool name and outcome this week — run from inside the ogcode chat to query the audit CSV without leaving the session.

```
$ query_sheet .ogcode/tool_calls.csv
SELECT tool, status, COUNT(*) FROM Sheet1 GROUP BY tool, status
```

— Gives a runnable query example against a named tool and file. v0.57.0

THINNER COVERAGE BELOW

03 ogcode-knowledge skill renamed and expanded

BREAKING

54

The built-in `ogcode-knowledge` skill (formerly `customize-ogcode`) now covers runtime state in `.ogcode/` and `~/.ogcode/`, the tool-call audit log, and a tour of the served UI. Any project skill named `customize-ogcode` intended to override the built-in will no longer shadow it after the rename.

— Names old/new skill IDs and breaking override behaviour, no fix steps given. v0.57.0

04 Agents page rebuilt as fixed-geometry board map IMPROVED

16

The Agents page has been rebuilt as a fixed-geometry board map, per the release summary; no further mechanism or UI detail is given.

— Only a bare summary mention, no detail on behaviour. v0.57.0

└──▷ BREAKING ON UPGRADE

! The built-in skill `customize-ogcode` is renamed to `ogcode-knowledge`; any project skill named `customize-ogcode` that was intended to override it will no longer shadow the built-in.

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

javis-os

Feedback ⓘ desktop NOTES ↗ RANK

> javis-os shipped its first native desktop installers for Windows and macOS, bundled with guided quickstart documentation.

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

01 **Native desktop installers for Windows and macOS** NEW

45

javis-os introduces desktop installers for Windows 10/11 and macOS 11+ as the first native-app distribution of the platform. After installing, users can follow `QUICKSTART.md` (Tiếng Việt) or `QUICKSTART.en.md` (English) for guided setup.

— Names OS versions and quickstart files but no mechanism detail desktop

WAS THIS USEFUL? 👍 👎

Graphicle AI Graphicle

Feedback ⓘ v0.1.11 NOTES ↗

RANK

Code graph and memory for your coding agents, in one Rust program. For Claude Code, Codex, Cursor and any MCP client.

> Graphicle's v0.1.11 release adds dynamic context-file injection to `graphicle fix` rounds, a new Fireworks benchmark profile, and several tuning changes to its Muse Code integration for output size and event parsing.

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

01 **Dynamic context files in graphicle fix rounds** NEW 88

After a failed round, `graphicle fix` now injects up to 6 files and 24,000 tokens named in test output, the model's explanation, or the code graph into the next round's request. These added files are listed under `context` in each round's JSON report and in the readable report; documented in `docs/fix.md`.

— Names exact limits, config surfaces and doc reference v0.1.11

02 **Fireworks benchmark profile for whole-repo bench** NEW 85

Adds a `fireworks` profile to `bench/whole-repo/run.py` for running Muse Spark Open on Fireworks' serverless API. It reads the API key from `FIREWORKS_API_KEY` or `--key-file`, sends session affinity hints via the `x-session-affinity` header and `user` field, and pulls cached-token counts from response usage or the `fireworks-cached-prompt-tokens` header, with prices read from `prices.json`.

— Full flag, header and file names given, runnable v0.1.11

03 **Muse Code 1.4.14 event field support in bench kit** IMPROVED 68

`bench/muse-code/` now reads Muse Code 1.4.14 event fields: tool call and token data come from the `run.usage` event, the answer from `run.terminal.completed`, and event names from `payload_type`, so the kit correctly counts calls and cost for 1.4.14 sessions.

— Names exact event fields but no user-facing command v0.1.11

04 **Session-start output cap in Muse Code plugin** IMPROVED 60

The Muse Code plugin now caps session-start output at 14,000 bytes to stay within Muse Code's 16,384-byte hook limit, dropping the least-connected files first, then

documentation, folder map, and architecture summary, then the memory's recent turns, and appending a line naming what was left out and which tool returns it.

– Describes mechanism and limits but not a direct command v0.1.11

THINNER COVERAGE BELOW

05 **Lower default session map token limits** IMPROVED 55

Muse Code's session map now defaults to 2,500 tokens and caps at 3,500 tokens, down from 8,000 and 100,000, listing the most connected files first.

– Clear before/after numbers but no config key named v0.1.11

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.8.231 NOTES ↗ RANK

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

> HyperFrames made Studio's embedded editor shortcuts fully remappable by host applications and added an API export for offloading heavy media uploads.

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

01 **offloadHeavyMedia API export** NEW 65

A new `offloadHeavyMedia` API export lets host apps upload project media themselves, returning an `asset_id` for every offloaded file.

– Names exact API and return field, but no usage example v0.8.227

THINNER COVERAGE BELOW

02 **Remappable Studio shortcuts for host apps** IMPROVED 35

Host applications embedding Studio can now remap its shortcuts: first single-key shortcuts became remappable, then this was extended so every Studio editor and player shortcut is remappable by the embedding host application.

– Describes scope expansion but no API/config names given v0.8.231 v0.8.230

WAS THIS USEFUL? 👍 👎

agent-app-orchestrator

Feedback ⓘ v7.0.0 NOTES ↗

RANK

Agent orchestration service for the Agent Landing Zone, built on Microsoft Foundry Agent Service and Microsoft Agent Framework. Coordinates specialized agents for conversational experiences with grounded knowledge, NL2SQL and MCP tools.

> agent-app-orchestrator moved to native Microsoft Agent Framework conversation and workflow modes with new host/mode routing, then replaced its direct NL2SQL connectors with generic, managed-identity-authenticated MCP tool discovery, retiring legacy strategy configuration and API endpoints along the way.

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

01 MCP tool discovery with managed-identity authentication NEW

95

Generic MCP tool discovery can be enabled with `MCP_ENABLED`, using invocation-scoped managed-identity bearer authentication via `MCP_AUTH_RESOURCE_SCOPE=api://<MCP API client ID>/.default`, refreshed before expiry without API-key fallback (external API-key MCP remains supported when no resource scope is set). Connectivity is configured with `MCP_APP_ENDPOINT` and `MCP_TRUSTED_HOSTS` (JSON trusted-host arrays) over the `streamable-http` transport, and MCP enablement, connection, and authentication settings are refreshed per native invocation, including after identity setup and resumed workflows. Server-side toolsets are controlled with `MCP_TOOLSETS`: setting `["nl2sql"]` enables the `nl2sql.list_sources` and `nl2sql.query` tools, while the default `[]` runs an authenticated MCP server with no tools and no NL2SQL initialization.

Enable MCP tool discovery with managed-identity bearer auth so the orchestrator authenticates to the MCP server without API keys, refreshing credentials automatically.

text

```
MCP_ENABLED=true
MCP_APP_ENDPOINT=https://<mcp-server-host>/mcp
MCP_TRUSTED_HOSTS=["<mcp-server-host>"]
MCP_AUTH_RESOURCE_SCOPE=api://<MCP API client ID>/.default
```

Restrict the MCP server to NL2SQL tools only for applications that need SQL access, while leaving non-SQL applications with a clean authenticated server and no NL2SQL initialization overhead.

```text
MCP_TOOLSETS=["nl2sql"]
```

— Full env-var config surface with runnable examples v7.0.0

02 **Orchestration host and application mode selection** NEW

90

Select the orchestration host by setting `ORCHESTRATION_HOST` to `hosted` or `container_apps`, and set `APPLICATION_MODE` to `conversational`, `workflow`, or `both` to control which execution paths are active. New `native.durable` and `native.hosted` modes run typed human-reviewed workflows or native hosted conversations, registered as `agent-lz-research-review` and `agent-lz-conversation` respectively.

Run the orchestrator in hosted conversational mode for a standard chat deployment.

```env
ORCHESTRATION_HOST=hosted APPLICATION_MODE=conversational
```

Enable both conversational and human-reviewed workflow modes when using Container Apps with separate publications for each.

```env
ORCHESTRATION_HOST=container_apps
APPLICATION_MODE=both
```

— Exact env vars and values with runnable examples v6.0.0

03 **Direct NL2SQL connectors removed** BREAKING

80

Direct NL2SQL registration and all SQL/Fabric/DAX connectors are removed in favor of MCP; the environment variables `NL2SQL_ENABLED`, `NL2SQL_SOURCE_REGISTRY`, `NL2SQL_QUERY_TIMEOUT_SECONDS`, `NL2SQL_MAX_ROWS`, `DATASOURCES_CONTAINER`, `NL2SQL_STORAGE_CONTAINER`, `SEARCH_TABLES_INDEX_NAME`, `SEARCH_MEASURES_INDEX_NAME`, `SEARCH_QUERIES_INDEX_NAME`, `SEARCH_TABLES_INDEX_TOP_K`, and `SEARCH_QUERIES_INDEX_TOP_K` no longer configure orchestrator SQL, and no legacy SQL or GPT-RAG MCP fallback is provided.

— Lists every retired env var for migration v7.0.0

04 **Legacy strategy and endpoints retired** BREAKING

70

The classic `AGENT_STRATEGY` environment variable is retired, requiring migration to the new native contracts; the `/orchestrator`, `/conversations`, and `/api/dashboard` endpoints are also retired, and any integrations calling them will break on upgrade.

— Names exact retired variable and endpoints v6.0.0

THINNER COVERAGE BELOW

05 **Structured NL2SQL results over MCP** IMPROVED 55

NL2SQL results returned over MCP now preserve schema-version-1 source, query and metadata IDs, typed columns, rows, and count, with text-only JSON support and explicit tool errors surfaced through the official MCP client.

— Describes format in detail but no invocation steps v7.0.0

06 **Native workflow infrastructure additions** IMPROVED 35

Adds explicit ManagedIdentity DTS binding, separate production host dependencies, optional NL2SQL and read-only SQL guards, and retrieval/MCP and attachment serialization.

— Thin list with no config keys or commands v6.0.0

└──▷ BREAKING ON UPGRADE

! Direct NL2SQL registration and all SQL/Fabric/DAX connectors are removed; the environment variables `NL2SQL_ENABLED`, `NL2SQL_SOURCE_REGISTRY`, `NL2SQL_QUERY_TIMEOUT_SECONDS`, `NL2SQL_MAX_ROWS`, `DATASOURCES_CONTAINER`, `NL2SQL_STORAGE_CONTAINER`, `SEARCH_TABLES_INDEX_NAME`, `SEARCH_MEASURES_INDEX_NAME`, `SEARCH_QUERIES_INDEX_NAME`, `SEARCH_TABLES_INDEX_TOP_K`, and `SEARCH_QUERIES_INDEX_TOP_K` no longer configure orchestrator SQL. No legacy SQL or GPT-RAG MCP fallback is provided.

! The classic `AGENT_STRATEGY` environment variable is retired; deployments relying on it must migrate to the new native contracts.

! The `/orchestrator`, `/conversations`, and `/api/dashboard` endpoints are retired; any integrations calling these endpoints will break on upgrade.

WAS THIS USEFUL?  

CrewAI

Feedback ⓘ 1.15.34 NOTES ↗ RANK

Open-source Python framework for building multi-agent workflows using role-based AI agent Crews and event-driven Flows.

> CrewAI's 1.15.34 release adds CLI-based skill discovery, runtime input passing for Python flows, and zip-based project deployment, while removing the agent executor introduced two releases prior.

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

01 **Zip-based project deployment** NEW 70

Build and deploy a project as a zip archive with `deploy create --zip` (invoked as `crewai deploy create --zip`), which also prints the commit the zip was built from for traceability.

Deploy a project as a zip bundle and capture the exact commit it was built from for audit or rollback.

```
$ crewai deploy create --zip
```

— Exact command plus commit-printing behavior specified. 1.15.34

02 **Skill search from the CLI** NEW 60

Run `crewai skill search` to search your organization's available skills directly from the CLI before wiring them into an agent.

Discover skills available in your organization before wiring them into an agent.

```
$ crewai skill search
```

— Names exact command but little on search mechanics or output. 1.15.34

03 **Runtime inputs for Python flows** NEW 60

Pass runtime inputs to a Python flow using `crewai run --inputs`, allowing dynamic values to be supplied without hardcoding them in the project script, e.g. `crewai run --inputs '{"topic": "supply chain risk", "depth": 3}'`.

Pass dynamic runtime values into a Python flow without hardcoding them in the project script.

```
$ crewai run --inputs '{"topic": "supply chain risk", "depth": 3}'
```

— Exact flag and example payload given, mechanism beyond that unspecified. 1.15.34

THINNER COVERAGE BELOW

04 **Agent executor removed** BREAKING 25

The agent executor introduced in 1.15.29 has been removed; migration notes are documented, though the specifics of the migration path are not included here.

— Notes removal but omits migration details or new path. 1.15.34

└──▷ **BREAKING ON UPGRADE**

! The agent executor introduced in 1.15.29 has been removed; migration notes are documented.

WAS THIS USEFUL? 👍 👎

cotal

Feedback ⓘ v0.133.13 NOTES ↗ RANK

The open standard for agent coordination

> This window's biggest addition is a trio of new `@cotal-ai/workspace` library exports covering managed launches, manager control calls and mesh viewing, alongside upfront channel-pattern validation in `cotal_send` (a breaking change for callers parsing old error text), named provider error codes for Jcode readiness failures, and reliability/display fixes in `cotal watch`.

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

01 Upfront channel-pattern validation in `cotal_send`

BREAKING

85

`cotal_send` now validates channel patterns before any read or send occurs: wildcard patterns such as `team.>` and malformed names such as `team.**` are refused immediately, with the refusal message explaining how to reach target agents — post to a matching channel to reach all connected followers at that moment, or post to each channel by name. In pi, this shared refusal replaces the old per-line error text, so callers that parsed it will need updating.

Catch an invalid wildcard channel pattern early before any message is sent or received.

```
$ cotal_send team.> 'hello'
```

— Exact patterns, mechanism and breaking-change note given v0.133.12

02 New `@cotal-ai/workspace` library exports

NEW

80

Adds `@cotal-ai/workspace/launch` with `requestManagedLaunch` and `readManagedLaunch` for sending a `ManagedLaunchRequest` as a manager `start` command and reading the joined agent as a `ManagedLaunchResult`. Adds `@cotal-ai/workspace/control` with `askManager`, `scatterManager`, `locateSeat`, `control`, `controlPs`, `deleteChannel`, `staleStoreHint` and `nativeEndpointRefusal` for manager control calls outside CLI code. Adds `@cotal-ai/workspace/view` with `MeshView` and a new `openObserver` helper that builds the read-only observer endpoint previously constructed inline by `cotal console` and `cotal watch`.

— Many named functions but no runnable usage sample given v0.133.13

03 Broker-outage status and thread fields in `cotal watch`

IMPROVED

75

`cotal watch` now reads `Reconnecting` during a broker outage instead of showing a false `live` status, and `cotal watch --json` rows now include `replyTo`, `mentions` and `contextId` fields when the sender set them.

Stream live mesh events as JSON and capture thread context fields for downstream correlation or alerting.

```
$ cotal watch --json | jq 'select(.replyTo != null) | {replyTo, mentions, contextId}'
```

— Clear before/after with exact command example `v0.133.13`

---

04 **Named provider error codes for Jcode readiness failures**

IMPROVED 60

Provider HTTP 403 and HTTP 429 responses during Jcode readiness are now named `provider_forbidden` and `provider_rate_limited` instead of a generic unknown startup failure, without exposing raw provider output.

— Named codes given but no command to trigger or inspect them `v0.133.12`

THINNER COVERAGE BELOW

05 **Terminal-visible documentation links** IMPROVED 20

Documentation link destinations are now visible in terminal output so readers can follow CLI reference URLs; piped, plain and JSON pages keep their Markdown.

— Thin description with no concrete mechanism or example `v0.133.13`

┗—▷ BREAKING ON UPGRADE

! In pi, `cotal_send` now returns the shared channel-pattern refusal instead of its own line when a wildcard pattern such as `team.>` or a malformed name such as `team.**` is supplied — callers that parsed the old per-line error text will need to be updated.

WAS THIS USEFUL? 👍 👎

octop-harness

Feedback ⓘ v1.0.6 NOTES ↗ RANK

A production-grade agent runtime engineered from the Harness Engineering philosophy.

> octop-harness v1.0.6 adds three built-in web search integrations and managed MCP sessions with HITL scope, alongside a breaking change to error handling in HttpDecisionClient.

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

01 HttpDecisionClient raises DecisionRequestError on failure

80

BREAKING

`HttpDecisionClient` now raises `DecisionRequestError` in non-strict mode after a transport or HTTP failure, instead of returning an empty result; callers that previously checked for empty results must now catch `DecisionRequestError`.

— Names exact class and exception plus required migration v1.0.6

THINNER COVERAGE BELOW

02 DeepSeek, Yuanbao, and You.com search integrations NEW

50

Adds DeepSeek Search and Yuanbao Search as built-in web search tools, plus an optional You.com search integration.

— Names three integrations but gives no activation or config details v1.0.6

03 Managed MCP sessions with HITL scope NEW

40

Adds managed MCP sessions that preserve HITL (human-in-the-loop) request scope.

— States the capability but not how sessions or scope work v1.0.6

04 Refreshed HAI and DeepSeek provider presets IMPROVED

30

Refreshes the HAI and DeepSeek provider preset catalogs.

— Bare mention of a catalog refresh with no specifics v1.0.6

└──▷ **BREAKING ON UPGRADE**

! `HttpDecisionClient` now raises `DecisionRequestError` (instead of returning an empty result) after a transport or HTTP failure in non-strict mode — callers that checked for empty results must now catch `DecisionRequestError`.

WAS THIS USEFUL? 👍 👎

◆ MCP Servers & Tooling

memwhale

Feedback ⓘ v0.25.0 NOTES ↗ RANK

Persistent, local memory for developers and their coding agents. Records commands, output, errors, and the fixes that worked into SQLite and serves them over MCP.

> memwhale ships search abstention so queries with no real matches return an explicit no-evidence signal instead of unrelated rows, alongside a new fix-verification field for checking whether past fixes still hold and a new retrieval method in its core library.

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

01 Search abstention when no content matches

BREAKING

80

Searches now filter rows to content matches before applying the result limit, returning an insufficient-evidence message instead of unrelated recent rows when nothing matches the query. This applies across `mw search`, MCP's `search_memory`, the dashboard search API, the TUI, and the desktop app. Scripts that expected rows from a search with no matching text now receive the insufficient-evidence message or an empty result instead.

Confirm that a query with no matching content returns an abstention message rather than unrelated rows.

```
$ mw search qqq-nothing-zzz
```

— Names every affected surface and the breaking behavior change, with a sample command.

v0.25.0

02 Fix verification via fix_candidates run argument

BREAKING

75

The `fix_candidates` function in `memorywhale-cli` now accepts a third argument — the run being hinted about (`i64::MAX` for none) — and `FixCandidate` gains a `verification` field, letting callers check whether a previously recorded fix still holds or has been invalidated by a later failed rerun. Callers that pass only two arguments to `fix_candidates` will break.

Check whether a previously recorded fix still holds, or has been invalidated by a later failed rerun.

```
$ mw hint 'error: linker `cc` not found'
```

— Names the function, new argument, and field, plus a usage command. v0.25.0

THINNER COVERAGE BELOW

03 **retrieve_matches method in memorywhale-core** NEW 45

The `memorywhale-core` library adds `MemoryEngine::retrieve_matches` with a default implementation, while `retrieve` remains unchanged.

— Names the new method but gives no behavior detail or example. v0.25.0

04 **Paired-trial scorer in release gate benchmarks** NEW 30

memwhale adds a paired-trial scorer in `benchmarks/release_gate/evaluate.py`.

— Only named in the summary with no further description. v0.25.0

└──▷ BREAKING ON UPGRADE

! Scripts that expected rows from a search with no matching text now receive the insufficient-evidence message or an empty result instead.

! `memorywhale-cli`'s `fix_candidates` function takes a third argument (the run being hinted about, `i64::MAX` for none); callers that pass only two arguments will break.

WAS THIS USEFUL? 👍 👎

kordoc

Feedback ⓘ v4.24.37 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's single release in this window adds a selective OCR re-recognition mode that reruns low-confidence pages with a handwriting model.

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

01 **Selective handwriting OCR re-recognition** NEW 93

Enable `--ocr-rec auto` (CLI) or `ocr_rec: "auto"` (MCP config) to automatically re-read pages where average OCR confidence falls below 0.9 using a handwriting model; a page's result is replaced only if the score improves by 0.06 or more, while printed pages stay on the default model.

Re-process a scanned PDF that mixes printed and handwritten pages, letting kordoc automatically apply the handwriting model only where it helps.

```
$ npx kordoc 스캔본.pdf --ocr --ocr-rec auto -o 스캔본.md
```

— Names exact flag, config key, and thresholds with runnable example. v4.24.37

WAS THIS USEFUL? 👍 👎

Composio

Feedback ⓘ @composio/cli@0.6.0 NOTES ↗ RANK

Composio connects AI agents to external tools and services with managed authentication and integrations.

> Composio's CLI 0.6.0 tightens account handling and process control: explicit account selection is now required in ambiguous cases, listener commands can fail fast on sink errors, and signal handling now returns structured exit codes instead of silently succeeding.

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

01 Explicit account selection for multi-account toolkits

BREAKING

80

Pass `--account <id|alias>` to disambiguate when a toolkit has multiple accounts and no `default` alias set. Toolkits with multiple accounts and no default alias no longer fall back to the newest account implicitly — callers must pass `--account <id|alias>` or assign a default alias, a behavior change from prior silent fallback.

— Names exact flag and describes before/after behavior change @composio/cli@0.6.0

02 Fail-fast listener exit on sink failure NEW

80

Add `--fail-fast` to `listen` and `dev triggers listen` so the listener exits immediately on sink failure instead of continuing silently, useful for surfacing pipeline failures loudly in CI.

Keep a trigger listener running in CI but halt immediately if any sink call fails, so pipeline failures are loud and fast.

```
$ composio listen --fail-fast
```

— Runnable command example with clear flag and behavior @composio/cli@0.6.0

03 Structured signal exit codes BREAKING

75

Commands interrupted by SIGINT, SIGTERM, or SIGHUP now exit with codes 130, 143, or 129 respectively; `run` forwards signals to its child process group and preserves the child's exit code. Scripts that previously treated this cancellation as success (exit 0) must now handle these specific exit codes.

— Exact codes and migration note but no command to run @composio/cli@0.6.0

THINNER COVERAGE BELOW

04 **Account metadata in connection listings** IMPROVED 45

Connection listings now include account IDs, aliases, word IDs, and provider display names when available.

— Names fields shown but no navigation or command given `@composio/cli@0.6.0`

05 **Resolvable project required for dev account linking** BREAKING 45

Developer account linking (`dev connected-accounts link`) now requires a resolvable developer project and acting user — initialize the working directory or explicitly select a project before linking.

— Names command and gives a clear migration starting point `@composio/cli@0.6.0`

06 **Telemetry exclusions for sensitive data** IMPROVED 40

Telemetry now excludes rejected flags, script-tail options, credentials, request bodies, and header values from lifecycle properties.

— Lists excluded data types but requires no reader action `@composio/cli@0.6.0`

└──▷ BREAKING ON UPGRADE

! Scripts that treated SIGINT/SIGTERM/SIGHUP cancellation as success (exit 0) must now handle exit codes 130, 143, and 129.

! Developer account linking (`dev connected-accounts link`) now requires a resolvable developer project and acting user — initialize the working directory or explicitly select a project before linking.

! Toolkits with multiple accounts and no `default` alias no longer fall back to the newest account implicitly; pass `--account <id|alias>` or assign a default alias.

WAS THIS USEFUL?  

xiaohongshu-mcp

Feedback ⓘ v2.7.0 NOTES ↗ RANK

MCP for xiaohongshu.com

> xiaohongshu-mcp added an original-content flag for video publishing and introduced optional noVNC Docker images published to dual registries.

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

01 **Original content flag for video publishing** NEW 60

Video posts can be marked as original content by passing `is_original` to the `publish_with_video` command.

— Names exact flag and command, no further mechanism given v2.7.0

THINNER COVERAGE BELOW

02 **Optional noVNC Docker image with dual-registry publishing** NEW 45

The MCP service can now be run with or without the noVNC browser UI by choosing between optional noVNC Docker images, which are published to dual registries.

— Describes option and registries but no image names or commands v2.6.0

WAS THIS USEFUL? 👍 👎

premiere-pro-mcp

Feedback ⓘ v1.28.0 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 an After Effects round-trip workflow, UXP clone-based sequence copying, and an audio signal check on direct exports, plus a config switch to force CEP-only bridge mode.

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

01 **Audio signal check on sequence export** NEW 85

`export_sequence audio_check=report|expect_signal` distinguishes first-stream decoded signal, digital silence, absent audio, and unavailable evidence, letting a workflow catch silent or missing audio before committing to a direct export.

Check whether a sequence's audio track actually carries a decoded signal before committing to a direct export — catch silent or missing audio early.

```
$ export_sequence audio_check=report
```

— Runnable command with exact flag values and distinguished states v1.28.0

02 **CEP-only bridge mode via environment variable** NEW 70

Setting `PREMIERE_BRIDGE_MODE=cep` withholds UXP tools and forces CEP bridge selection when UXP is unavailable or unwanted.

Force the server to use only the CEP bridge on a host where the UXP panel is not available or not authenticated.

```
$ PREMIERE_BRIDGE_MODE=cep npx premiere-pro-mcp
```

— Named env var with runnable example command v1.28.0

03 **After Effects round-trip composition** NEW 70

Preview and confirm an After Effects round trip that creates a composition from a timeline video clip, imports it over Dynamic Link, and replaces the original video span while retaining linked audio, with pre-apply disclosure of effects that will not carry over.

— Mechanism detailed but no exact invoking command named v1.28.0

04 **UXP clone-based sequence copying** NEW 65

Copy a clip range between sequences using UXP clone actions with inspect/apply guards, edge trimming in a parking area, explicit Undo counts, and structural readback.

— Mechanism described but no named command or flag given `v1.28.0`

WAS THIS USEFUL? 👍 👎

embody

Feedback ⓘ `v6.3.3` NOTES ↗ RANK

The most comprehensive MCP toolkit for TouchDesigner

> Embody's v6.3.3 release overhauls Convoy with a persistent, per-computer machine model that keeps TouchDesigner sessions reachable without the app open.

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

01 **Convoy Machines replace Convoy Nodes** BREAKING 76

The `convoy_list_nodes` MCP tool gains a `machines` field, replacing the old Convoy Nodes model with Convoy Machines: one row per computer showing `Hostname / IP`, its open `.toe` files, and `Online / 5s ago` status. Convoy now keeps a computer reachable at every login without TouchDesigner open, so an AI session can reach any machine in the room persistently after a one-time setup, and projects on a joined machine register automatically unless the Enable Convoy setting is turned off for that project.

— Names the exact field, model change, and config toggle. `v6.3.3`

└──▷ **BREAKING ON UPGRADE**

! Convoy Nodes is renamed to Convoy Machines; any code or tooling that references the old Convoy Nodes row model must be updated to the new per-computer Convoy Machines layout.

WAS THIS USEFUL? 👍 👎

brightdata-mcp

Feedback ⓘ v4.2.0 NOTES ↗ RANK

A powerful Model Context Protocol (MCP) server that provides an all-in-one solution for public web access.

> v4.2.0 reworks brightdata-mcp's core scraping tools end to end: `scrape_as_markdown` and `scrape_batch` gain a selectable output format, structured JSON responses, enforced timeouts, standardized error codes, and a switch from Web Unlocker to Bright Data's Fetch API that drops JavaScript rendering and caps page content length.

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

01 **Output format selection for scrape tools** NEW 83

`scrape_as_markdown` and `scrape_batch` gained a `format` parameter with values `markdown` (default), `text`, `html`, or `ld_json`, letting callers choose plain text, raw HTML, or JSON-LD metadata instead of rendered markdown.

Extract structured data from a page as LD+JSON to pull schema.org metadata without parsing HTML.

```text
scrape_as_markdown(url='https://example.com/product/123',
format='ld_json')
```

Scrape multiple URLs at once and get plain text back, useful for feeding pages into an LLM context window without markup noise.

```text
scrape_batch(urls=['https://example.com/a',
'https://example.com/b'], format='text')
```

— Named param and values with two runnable example calls v4.2.0

02 **Structured JSON responses for scrape tools** BREAKING

73

`scrape_as_markdown` and `scrape_batch` now return structured JSON objects with `url`, `title`, `format`, `content`, plus `final_url` after a redirect and a `note` field when relevant, instead of plain text. `scrape_batch` returns results as an array in input order, with failed URLs reported as `{url, error}`; this breaks any integration expecting a raw string response.

— Names exact response fields but no migration example v4.2.0

03 **Timeout handling for scrape requests** NEW

70

`scrape_as_markdown` now enforces a 45-second timeout and returns a `timeout` error when exceeded; `scrape_batch` cancels at 35 seconds, returning completed pages while marking remaining URLs as `timeout` errors.

— Specific timeout values and resulting error behaviour named v4.2.0

04 **Content length caps on scraped pages** BREAKING

68

Page content is now capped at 200,000 characters for `scrape_as_markdown` and 100,000 characters per page for `scrape_batch`, with a `note` field returned when truncation occurs; pipelines relying on full-page content may receive incomplete results.

— Exact character limits and truncation signal named v4.2.0

05 **Standardized error codes for scrape failures** NEW

63

Scrape failures now return clear one-line error codes: `page_not_found`, `fetch_failed`, `invalid_url`, `rate_limited`, `empty_page`, `unsupported_content`, `no_data`, `timeout`, `auth_failed`, `busy`, `service_error`, and `network_error`.

— All eleven error codes listed, but no usage example v4.2.0

THINNER COVERAGE BELOW

06 **Fetch API replaces Web Unlocker backend** BREAKING

53

`scrape_as_markdown` and `scrape_batch` now use Bright Data's Fetch API instead of Web Unlocker, with no JavaScript rendering — pages that previously relied on JS rendering will no longer render correctly.

— States the swap and its consequence but no remediation path v4.2.0

└──▷ **BREAKING ON UPGRADE**

! `scrape_as_markdown` and `scrape_batch` now use Bright Data's Fetch API instead of Web Unlocker — pages requiring JavaScript rendering will no longer render correctly.

! Both `scrape_as_markdown` and `scrape_batch` now return JSON objects instead of plain text, which will break any integration that expects a raw string response.

! Page content is truncated at 200,000 characters (`scrape_as_markdown`) or 100,000 characters per page (`scrape_batch`); pipelines relying on full-page content may receive incomplete results.

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ AI / LLM Gateways

dario

Feedback ⓘ v6.20.0 NOTES ↗ RANK

One local endpoint that routes Cursor, Cline, Aider, Claude Code and the Agent SDK through your Claude and ChatGPT subscriptions, with failover and pooling and zero runtime dependencies.

> dario now tracks thinking-token spend and lets operators pin a default reasoning effort per API key, with CLI and admin API surfaces for both plus a clear precedence rule for resolving effort across clients.

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

01 Default effort level on named keys NEW

95

A named key can now carry a default effort: set it with `dario keys create <name> --effort=low` or `dario keys effort <name> low`, and remove it with `--clear`. `dario keys list` gains an `EFFORT` column and `dario keys --json` carries an `effort` field; the same can be done over the admin API via `POST /admin/keys` (with an `effort` field) or `POST /admin/keys/<name>/effort`, and `GET /admin/keys` now returns `effort` too.

Create a key that caps all client requests to low effort by default, without requiring every client to set it themselves.

```
$ dario keys create ci-runner --effort=low
```

Downgrade an existing key to low effort, or remove the default to let the proxy fallback apply.

```
$ dario keys effort my-key low
# or remove it:
dario keys effort my-key --clear
```

— Lists CLI commands and admin endpoints verbatim. v6.20.0

02 Thinking-token cost tracking NEW

93

`dario usage` reads `output_tokens_details.thinking_tokens` from upstream for both streaming and buffered responses, adds a `Thinking:` line showing share of output tokens and cost at output rates, and can break this down per key with `--by-key`. The `lifetime` object in `/analytics` now also carries `thinkingCost` and `tokens.thinking` fields.

Check how much of your Opus spend is going to thinking tokens before deciding to switch models.

```
$ dario usage --by-key
```

— Names exact fields and flag, fully actionable. v6.20.0

03 **Effort resolution precedence** IMPROVED 65

Effort now resolves through a defined order: a model-name suffix first, then an `--effort` flag or `DARIO_EFFORT` pin, then the client's own field (`output_config.effort`, `reasoning_effort`, or `reasoning.effort`), then the key's default effort, and finally `high` as the fallback.

— Explains mechanism but no direct command to invoke it. v6.20.0

WAS THIS USEFUL? 👍 👎

◆ Inference Serving & Optimization

Experiential Labs Experiential

Feedback ⓘ v0.7.226

NOTES ↗ RANK

Experiential is the open source, zero markup gateway for BYOK, self-hosted and 1000+ marketplace models. It learns from your traffic to cut costs, recommend better models, and train a specialized model you own.

> Experiential's v0.7.226 release adds visibility into decision-model scoring and extends tag-based routing to project aliases.

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

01 **Decision-model category scores on checks** NEW 20

Experiential now shows decision-model category scores for every answered check.

— Names the feature but no mechanism or UI path given. v0.7.226

02 **Project alias routing by decision-model tags** NEW 20

Experiential now routes project aliases by decision-model tags.

— States the capability only; no config or command detail. v0.7.226

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

MLX Serve

Feedback ⓘ v26.11.1 NOTES ↗

The mlx-serve runtime provides Apple Silicon inference with OpenAI- and Anthropic-compatible APIs, chat, agent mode, and tool calling.

mlx-serve adds a plugin system, a chat usage dashboard, stream timing on OpenAI and Anthropic APIs, and `MLX_SERVE_F16_PREFILL=1` for faster M1/M2 prefill.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v26.11.1 https://github.com/ddalcu/mlx-serve.git
# already have the repo? check out this version:
$ git checkout v26.11.1
```

NEW SUPPORTED PLATFORM Scaffold, test and install model plugins with `mlx-serve plugin new`, `plugin check`, and `plugin install`; plugins appear under Models ▸ Plugins with an on/off switch, dependency view, and one-click pack download (experimental).

Enable faster prompt processing for bf16 models on M1/M2 Macs with the `MLX_SERVE_F16_PREFILL=1` environment variable.

Use `mlx-serve launch codex` or the Codex tab to run Codex against local models.

Stream timing data now appears in the final chunk of every streamed reply on both the OpenAI and Anthropic APIs.

A usage dashboard in the chat sidebar shows live Mac load and model speed with daily and 30-day history, per-model breakdown, persisting across restarts.

+2 more

NEW INTEGRATION NVIDIA Nemotron 3.5 Transcribe

NEW INTEGRATION Idol 3D model generation

NEW INTEGRATION Liquid LFM3 multimodal models

└──▷ BREAKING ON UPGRADE

! DiffusionGemma is no longer supported.

! Many `MLX_SERVE_*` tuning environment variables have been removed; the behaviours they enabled are now the default.

WAS THIS USEFUL? 👍 👎

◆ Agentic Infrastructure

Heretic

Feedback ⓘ v2.0.0 NOTES ↗ RANK

Heretic removes safety alignment from language models by tuning directional-ablation parameters.

> Heretic 2.0 introduces a generic plugin system for scoring and model modification, adds headless operation and an alternate `config.nosyco.toml` config path, and ships two breaking changes — incompatible v1.x configs and removal of research features in favor of Lophius.

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

01 **Plugin system for scorers and modifiers** NEW 76

Heretic adds a generic plugin system for custom scorer integrations, including a benchmark scorer to score abliteration quality against standard benchmarks, a Generic BERT Classifier scorer plugin to evaluate model outputs with a BERT-based classifier, and modifier plugins with ARA. Plugins can be marked as deterministic using the `reproducible` property on plugin definitions, and evaluations can now specify a dataset's specific config or subset.

– Names several plugin types and a property but no commands v2.0.0

02 **Disable MLP ablation via max_weight floor** NEW 72

The optimizer can now disable MLP ablation entirely by setting a `0` `max_weight` floor.

– Exact config value and effect named v2.0.0

THINNER COVERAGE BELOW

03 **Incompatible v1.x configuration files** BREAKING 58

Configuration files made for Heretic 1.x are not compatible with Heretic 2.0; a migration guide (docs/migration_v1_to_v2.md) explains how to update existing configurations.

– Links a migration guide but no detail of changed keys v2.0.0

04 **Alternate config.nosyco.toml config path** NEW 56

Heretic adds `config.nosyco.toml` as an alternative configuration file path.

– Exact config filename given, no further behaviour explained v2.0.0

05 **Headless operation mode** NEW 45

Heretic can now run without a UI using headless operation, now supported with end-to-end tests.

— States the capability but no invocation details v2.0.0

06 **Research features removed, replaced by Lophius** BREAKING

40

Research features have been removed from Heretic; the intended replacement is Lophius (`https://lophius.org/`).

— States removal and replacement, no mechanism detail v2.0.0

└──▷ **BREAKING ON UPGRADE**

! Configuration files made for Heretic 1.x are not compatible with Heretic 2.0 — see the migration guide at `https://github.com/p-e-w/heretic/blob/master/docs/migration_v1_to_v2.md` to update existing configurations.

! Research features have been removed; the intended replacement is Lophius (`https://lophius.org/`).

WAS THIS USEFUL? 👍 👎

◆ **Local LLM Hardware**

Apple Mac Studio

Feedback ⓘ hw-option-available-20261029

STORE ↗

Apple's desktop workstation with up to 512 GB of unified memory shared between CPU and GPU — the largest-memory single-box option for running open-weight models locally.

512GB memory option for M5 Ultra is now available at `www.apple.com` , first seen 29 Oct 2026.

WAS THIS USEFUL? 👍 👎

Minisforum MS-S1 Max

Feedback ⓘ shipped · Sep 20, 2026

STORE ↗

A Ryzen AI Max+ 395 mini workstation with 80 Gbps USB4 v2 and 2U rack-mount support, aimed at multi-unit local AI clusters.

Ryzen AI Max+ 395 began shipping on 20 Sep 2026.

WAS THIS USEFUL? 👍 👎

Beelink GTR9 Pro

Feedback ⓘ shipped · Sep 19, 2026 STORE ↗

A Mac Studio-styled mini PC on AMD's Ryzen AI Max+ 395, with dual 10GbE ports aimed at running it as a local AI compute hub.

Ryzen AI Max+ 395 began shipping on 19 Sep 2026.

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

stashbase

Feedback ⓘ v2.18.5 NOTES ↗ RANK

stashbase is an open-source, local-first writing workspace where you write with Claude Code and Codex, drawing on your own sources.

> Stashbase v2.18.5 adds consistent inline TeX formula rendering across Chat and Documents, lets users copy formatted replies and formulas from Chat into Documents, and introduces collapsible process messages in Chat.

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

01 **Inline TeX formulas via project Math toggle** NEW 71

A project-level Math toggle activates `$...$` inline formulas across Chat and Documents, and automatically updates Agent instructions for both replies and file writes, taking effect on the next turn in existing chats.

Turn on TeX formula rendering for a project so that inline math in Chat and Documents renders consistently.

📍 Open your project settings, locate the Math toggle, and enable it. Inline `$...$` formulas will now render in Chat replies and document file writes, and Agent instructions will be updated automatically.

— Named toggle and syntax, but only a UI path, no limits given v2.18.5

02 **Copy formatted Chat content into Documents** NEW 66

A reply, mixed passage, or selected formula can be copied from Chat into Documents while preserving Markdown and TeX, including matrix row separators, with formulas remaining editable after saving and reopening.

Copy a Chat reply containing formulas into a Document without losing formatting.

📍 In Chat, select the reply or formula you want to reuse, then use the copy action. Open or create a Document and paste — Markdown and TeX, including matrix row separators, are preserved and the formulas remain editable.

— Step-by-step copy workflow given with concrete preserved elements v2.18.5

THINNER COVERAGE BELOW

03 **Direct formula editing in Documents** IMPROVED 51

Users can click a formula in editing mode to edit its source directly, without opening the source or PREVIEW panels, with tighter block spacing and improved wrapping for long

source lines.

— UI interaction described but no exact navigation path given v2.18.5

04 **Collapsible process messages in Chat** NEW 44

Lengthy process messages fold automatically after successful Chat turns while final replies and file results stay visible, with an option to expand the process and a single contextual action in the composer while an Agent is running.

— Behavior described but no exact UI control or command named v2.18.5

WAS THIS USEFUL? 👍 👎

Volcengine OpenViking

Feedback ⓘ v0.5.3 NOTES ↗ RANK

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

> OpenViking v0.5.3 ships a Beta Rules service for agent rule search, a multi-replica Redis-backed Gateway with configurable session storage, retrieval tuning via rerank and chunk-size controls, self-service API key rotation, several new agent connectors, and a batch group-membership endpoint — alongside breaking changes to session URIs, the admin migrate route, and identity validation.

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

01 Batch group membership endpoint

BREAKING — 85

Admins can add group members in batch via the updated `PUT /api/v1/admin/accounts/{account_id}/groups/{group_id}/members` endpoint, accepting `{"user_ids": [...]}` with 1–1000 users; this replaces the removed `PUT /members/{user_id}` single-user route (send a one-element array for a single user).

Add multiple users to a group in one call using the new batch members endpoint.

```
$ curl -X PUT
https://<host>/api/v1/admin/accounts/<account_id>/groups/<group_
id>/members \
  -H 'Content-Type: application/json' \
  -d '{"user_ids": ["user-1", "user-2", "user-3"]}'
```

— Full endpoint, payload shape, and runnable curl example v0.5.3

02 Multi-replica Gateway with Redis-backed session storage

IMPROVED — 75

The standalone Gateway can now run as a multi-replica service on Redis, with configurable session expiry (in seconds), embedded gateway data stored under `storage.workspace`, and an opt-in Redis directory TTL for storage.

— Names config keys and mechanism but no exact values shown v0.5.3

03 **Retrieval tuning: rerank, chunk size, concurrency bypass**

IMPROVED 70

Retrieval is improved with opt-in rerank in context mode via the decoupled `rerank.threshold` config key, configurable L2 chunk size, and online retrieval now bypasses ingestion concurrency limits.

— Names config key and behavior change, no worked example v0.5.3

04 **CLI resource polling and progress display** IMPROVED 70

The CLI can poll task status after resource addition with `add-resource --wait`, which now also shows OVPack progress.

Wait for a resource ingestion job to complete before proceeding in a CI pipeline.

```
$ ov add-resource --wait <resource_path>
```

— Exact flag plus a runnable example command v0.5.3

05 **Session URIs restricted to per-user namespace** BREAKING 65

Sessions are now only read and written under `viking://user/{user_id}/sessions/...`; legacy `viking://session/...` URIs return `INVALID_URI`, and sessions existing only in legacy directories are treated as missing.

— Exact URI patterns and error code given, no migration tool named v0.5.3

06 **Admin migrate endpoint and CLI command removed** BREAKING 65

`POST /api/v1/admin/migrate` now returns `400 INVALID_ARGUMENT`; `ov admin migrate` and the matching Python, TypeScript, and Go SDK methods are removed. Deployments with 0.3.x sessions must migrate on an older release before upgrading.

— Exact endpoint, error code, and upgrade note but no replacement path v0.5.3

THINNER COVERAGE BELOW

07 **Rules (Beta) service for agent rule search** NEW 55

Agents can search rules via the new `openviking_rule_search` tool exposed by the new Rule service (Beta), backed by English and Chinese guides and an API reference.

— Names the new tool but gives no usage example v0.5.3

08 **New agent connectors: CodeBuddy, DeepSeek Harness, DingTalk** NEW 50

OpenViking adds a CodeBuddy Code CLI adapter, a DeepSeek Harness Desktop integration, and DingTalk connector OAuth support.

— Names three integrations with no setup detail v0.5.3

09 **Compile supports image sources with stricter namespace checks** IMPROVED 30

Compile now supports image sources and rejects system-namespace inputs and outputs.

— States the change but no flag or command given v0.5.3

10 **Stricter identity validation** BREAKING 30

Unknown roles and missing registered identities are now rejected by stricter identity validation.

— States the restriction with no error code or scope detail v0.5.3

11 **Self-service API key rotation** NEW 25

Users can now rotate their own API keys without admin involvement.

— Bare statement with no mechanism or location named v0.5.3

12 **Post-install onboarding and history import** NEW 20

Post-install agent onboarding and a history import script are now available.

— Thin description with no script name or steps v0.5.3

13 **Filesystem mutation commands in Studio terminal** NEW 20

The terminal now supports filesystem mutation commands in Studio.

— No commands or paths named v0.5.3

└──▷ BREAKING ON UPGRADE

! Sessions are now only read and written under `viking://user/{user_id}/sessions/...`; legacy `viking://session/...` URIs return `INVALID_URI` and sessions that exist only in legacy directories are treated as missing.

! `POST /api/v1/admin/migrate` now returns `400 INVALID_ARGUMENT`; `ov admin migrate` and the matching Python, TypeScript, and Go SDK methods are removed. Deployments with 0.3.x sessions must migrate on an older release before upgrading.

! The old `PUT /members/{user_id}` group-member route is removed; use `PUT /api/v1/admin/accounts/{account_id}/groups/{group_id}/members` with

`{"user_ids": [...]}` — for a single user, send a one-element array.

! Unknown roles and missing registered identities are now rejected by stricter identity validation.

WAS THIS USEFUL? 👍 👎

1c-ai-connector

Feedback ⓘ v1.9.1 NOTES ↗ RANK

ИИкона — AI-коннектор для 1С:Предприятие 8.3: LLM-провайдеры, агентская петля (function calling), RAG, MCP-сервер, мониторинг ошибок с ИИ-анализом, генератор диаграмм

> 1c-ai-connector's v1.9.1 release focuses on dialog management, adding bulk conversation deletion with atomic validation, per-conversation tabs, and small chat UI polish.

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

01 Bulk deletion of conversations in 'My Dialogs' NEW

72

Users can delete one or more selected conversations from the My Dialogs list via a single confirmation prompt that shows the count of selected conversations and defaults to No. Atomic deletion enforces all-or-nothing semantics: the whole selection is validated and locked before any removal, blocking the operation if any conversation is foreign, stale, has an in-progress request, or shares materials with another conversation. After successful deletion the list refreshes immediately, tabs for deleted conversations close, and drafts, history, and attachments are removed while the technical request log is preserved.

– Describes mechanism and edge cases but no exact UI path or command v1.9.1

THINNER COVERAGE BELOW

02 Per-conversation tabs in 1C NEW

52

Each conversation now opens as its own separate 1C tab with its own input field, draft, history, expert, model, and materials; closing a tab saves the conversation for later retrieval via My Dialogs.

– Clear behaviour change but no config or command detail v1.9.1

03 Compact chat commands with tooltips IMPROVED

23

Chat now shows compact commands using standard 1C icons, with tooltips on frequently used actions.

– Thin UI tweak with no further specifics v1.9.1

WAS THIS USEFUL? 👍 👎

rabitq-library

Feedback ⓘ v0.8.0 NOTES ↗ RANK

An official lightweight library for the RaBitQ algorithm and its applications in vector search.

> rabitq-library's v0.8.0 release brings full index-building workflows to Python for IVF, HNSW and the newly exposed SymqgIndex, adds flexible k-means clustering and deterministic quantization internals, and renames query factor accessors as a breaking change.

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

01 Centroid training built into IVF and HNSW indexes NEW

80

Python `train`, `set_centroids`, and `add` methods were added to `IvfIndex` and `HnswIndex`, letting indexes be built incrementally without pre-computing centroids externally. Centroid training now happens directly at the C++ level inside IVF and HNSW indexes, removing the requirement to supply pre-trained centroids separately.

Train centroids directly on an `IvfIndex` without a separate `RaBitQKMeans` step, then add vectors incrementally.

python

```python
from rabitqlib import IvfIndex
import numpy as np

rng = np.random.default_rng(1)
data = rng.standard_normal((500, 64)).astype(np.float32)

index = IvfIndex(dim=64, max_elements=len(data),
num_clusters=10, nbits=4, metric='l2')
index.train(data)
index.add(data)
ids, distances = index.search(data[:5], k=10, nprobe=5)
```

— Names methods and classes with a runnable example v0.8.0

02 SymqgIndex support in Python NEW

70

`train`, `add`, and `search` methods were added to `SymqgIndex` in Python, enabling full SymphonyQG index workflows to be run entirely from Python for the first time.

Build and search a SymphonyQG index entirely from Python — previously unavailable via the Python API.

```python
from rabitqlib import SymqgIndex
import numpy as np

rng = np.random.default_rng(0)
data = rng.standard_normal((1000, 64)).astype(np.float32)
queries = rng.standard_normal((10, 64)).astype(np.float32)

index = SymqgIndex(dim=64, max_elements=len(data), metric='l2')
index.train(data)
index.add(data)
ids, distances = index.search(queries, k=10)
```

— Named methods plus a runnable code example v0.8.0

THINNER COVERAGE BELOW

03 **Flexible clustering in RaBitQKMeans** IMPROVED 40

Optional k-means sampling and assignment of new vectors were added in `RaBitQKMeans` for more flexible clustering workflows.

— Names the class but no usage shown v0.8.0

04 **Deterministic quantization core** NEW 35

Portable seeded rotations and a frozen fast-mode scale table were added to the core, giving reproducible, deterministic quantization.

— Describes mechanism but no API surface or example v0.8.0

05 **Query factor accessors renamed with g_ prefix** BREAKING 30

Query factor accessors are renamed with the `g_` prefix, a breaking change for code relying on the previous accessor names.

— Names the change but no migration detail given v0.8.0

└──▷ BREAKING ON UPGRADE

! Query factor accessors are renamed with the `g_` prefix (refactor!: rename query factor accessors with the g_ prefix).

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ updated · Nov 2, 2026

NOTES ↗

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

LangSmith adds MCP tools for querying runs, managing prompts, and operating sandboxes; sandbox proxy credential routing via `match_headers`; LSU-based Engine spend display; and expanded deployment log detail and audit logging for Managed Tools.

UPDATED SUPPORTED PLATFORM 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, and returns token counts and USD costs by default.

The LangSmith MCP's `get_prompt_by_name` now returns the prompt template.

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 credential and serve multiple accounts.

+10 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

WAS THIS USEFUL? 👍 👎

Superlog Labs Superlog

Feedback

changes since 2026-10-03 CODE ↗

Superlog monitors AI agent telemetry with OpenTelemetry and groups signals into incidents for observability.

Superlog adds per-project Linear team selection, ClickHouse insert pacing controls, and GCP Pub/Sub connection caching to cut Postgres load

└──▷ TRY IT

Fetch all Linear teams available to a project so you can present them in a setup flow or verify the configured team ID before patching.

```
$ curl -s -H 'Authorization: Bearer <token>'
https://api.superlog.io/api/projects/<projectId>/linear/teams
```

Pin a project's agent tickets to a specific Linear team, or clear the override to fall back to the workspace's first team.

```
$ curl -s -X PATCH -H 'Authorization: Bearer <token>' -H 'Content-
Type: application/json' -d '{"linearDefaultTeamId": "<teamId>"}'
https://api.superlog.io/api/projects/<projectId>/automation
```

Control ClickHouse insert pacing with `INGEST_CLICKHOUSE_MIN_INSERT_INTERVAL_MS` to enforce a minimum gap between inserts into the same table per worker, growing batch sizes instead of producing more parts when inserts are fast.

Disable ClickHouse keep-alive with `INGEST_CLICKHOUSE_KEEP_ALIVE=false` to open one connection per insert and spread writes evenly across replicas instead of accumulating on the slowest one.

A new `GET /api/projects/:projectId/linear/teams` endpoint lists all teams in a project's connected Linear workspace, returning `{ installed: false, teams: [] }` when no connection exists and 502 when Linear is unreachable.

A Default team picker in Settings under the File Linear ticket step lets each project choose its Linear team, with Workspace's first team as the fallback option.

GCP Pub/Sub connection lookups are now cached for 60 seconds and concurrent misses collapse into one Postgres query, keeping the `/gcp/pubsub/:connectionId` route off the database for repeated pushes; unknown or non-ingestible connections are cached for 30 seconds.

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI Red Teaming & Testing

NVIDIA SkillSpector

Feedback ⓘ v2.13.3 NOTES ↗ RANK

SkillSpector scans AI agent skills for vulnerabilities and malicious patterns before installation.

> SkillSpector's latest release adds configurable scan limits via environment variables, broadens malware and credential-exfiltration detection across MCP configs, Pipenv dependencies, and urllib taint paths, and tightens report-destination safety with a breaking change to prevent overwriting input files.

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

01 Configurable scan limits via environment variables NEW

95

`SKILLSPECTOR_TP4_MAX_BATCHES` (default 256) controls the tool-description comparison batch limit, `SKILLSPECTOR_MAX_WORKFLOW_SECONDS` (default 1,080 seconds) sets the workflow deadline, and the existing `SKILLSPECTOR_MAX_LLM_CONCURRENCY` bounds parallel batch execution.

Run a large LLM-assisted scan that would previously time out under the default 18-minute deadline, raising the ceiling for big skills.

```
$ SKILLSPECTOR_TP4_MAX_BATCHES=512
SKILLSPECTOR_MAX_WORKFLOW_SECONDS=3600
SKILLSPECTOR_MAX_LLM_CONCURRENCY=8 skillspector scan ./large-
skill/
```

— Names exact env vars with defaults and a runnable command v2.13.3

02 Non-English gap-fill findings affect batch-scan verdicts

IMPROVED

65

Non-English gap-fill findings are now included in contributed batch-scan scores, recommendations, and exit status instead of retaining a stale SAFE verdict.

Scan a directory of skills in batch and get non-English gap-fill findings correctly reflected in scores and exit status.

```
$ python -m contrib.batch_scan.batch_scan ./my-skills/ --workers
20 -f json -o report.json
```

— Has a runnable batch-scan command demonstrating the fix v2.13.3

03 **Report destination overwrite protection** BREAKING 60

Baseline and recursive report destinations that would overwrite the primary input or a discovered child skill's input, including file aliases, are now rejected; existing scripts that write reports to the same path as the input will fail.

— Clear before/after and migration impact, no exact error message v2.13.3

THINNER COVERAGE BELOW

04 **Expanded MCP and agent credential file detection** IMPROVED 55

SkillSpector inspects `npx` commands in JSON MCP configurations and now recognizes additional Claude, Cline, and Roo Code credential and configuration filenames.

— Names tools and command but no runnable example v2.13.3

05 **Lower-confidence ransomware YARA signal** NEW 55

A new bounded, lower-confidence YARA signal flags nearby directory walking, encryption, and binary writes across lines to surface ransomware-like patterns, supplementing the existing stronger ransomware rules.

— Describes detection pattern but not a named rule or flag v2.13.3

06 **Taint tracking for credentials in urllib headers** IMPROVED 55

SkillSpector now follows credential reads nested in `urllib` request headers into network sinks for taint tracking.

— Names the library and mechanism, no usage example v2.13.3

07 **Pipenv dependency extraction and version checks** NEW 50

SkillSpector extracts Pipenv dependencies and checks locked versions, while preserving incomplete coverage for malformed or unsupported dependency declarations.

— Clear mechanism but no exact flag or output format v2.13.3

08 **Agent-memory file write detection from test carriers** NEW 45

SkillSpector now detects writes to known agent-memory files and directories originating from automatically run test carriers, including raw Windows paths.

— Describes behavior but no config surface or example v2.13.3

09 **Incomplete-inspection reporting for missing YARA rule directories**

IMPROVED

40

SkillSpector reports incomplete inspection when a requested custom YARA rule directory is unavailable.

— Thin prose description, no flag or path named v2.13.3

└──▷ BREAKING ON UPGRADE

! Baseline and recursive report destinations that would overwrite the primary input or a discovered child skill's input, including file aliases, are now rejected — existing scripts that write reports to the same path as the input will fail.

WAS THIS USEFUL? 👍 👎

◆ AI/LLM Security

containarium

Feedback ⓘ v0.125.0 NOTES ↗ RANK

Open-source agent runtime — SSH-native isolation, eBPF egress policy, Kubernetes + LXC backends, GPU passthrough, MCP-native CLI

> Containarium's v0.125.0 release strengthens disaster-recovery capabilities with whole-box backup verify/restore and automatic in-process recovery for several internal stores after a degraded start.

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

01 **Whole-box backup verify and restore** NEW

45

A whole-box backup can now be verified by restoring it into a scratch box, and a whole-box backup can be restored into a new box with the restore streamed and bounded.

— Describes mechanism but no exact command or flag given v0.125.0

02 **In-process recovery after degraded start** IMPROVED

45

Alert, credential-watch, passthrough, and sync-job stores now install in-process after a degraded start, rather than requiring a full restart.

— Names affected stores but no operational steps shown v0.125.0

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Open-Weight Models

Meituan LongCat

Feedback ⓘ LongCat-2.5 Now Available NOTES ↗

RANK

Meituan LongCat is a family of large language models from Meituan, offered via longcat.chat for chat and developer API access.

> Meituan shipped LongCat-2.5, adding native multimodality, long-horizon agentic coding for full codebases, and compatibility with popular coding agent harnesses.

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

01 Compatibility with Claude Code, Codex, Hermes harnesses NEW

60

LongCat-2.5 works across Claude Code, Codex, Hermes, and other agent harnesses, enabling flexible integration into development workflows.

— Names specific harnesses but gives no setup steps LongCat-2.5 Now Available

THINNER COVERAGE BELOW

02 Long-horizon agentic coding for full codebases IMPROVED

48

LongCat-2.5 now handles complex codebases and full-project builds through multi-step reasoning, tool use, and end-to-end task execution.

— Describes mechanism but no concrete limits or commands LongCat-2.5 Now Available

03 Native multimodality for visual coding and design NEW

38

LongCat-2.5 now combines visual understanding with agentic coding to support visual coding, creative design, and 3D tasks.

— States capability but no mechanism or usage detail LongCat-2.5 Now Available

WAS THIS USEFUL? 👍 👎

H2O.ai h2o-lightning-bolt-74b

Feedback

released · Nov 2, 2026

MODEL ↗

h2o-lightning-bolt-74b from H2O.ai was released on 1 Nov 2026. Open weights on Hugging Face.

WAS THIS USEFUL?

H2O.ai h2o-lightning-e4b

Feedback

released · Nov 2, 2026

MODEL ↗

h2o-lightning-e4b from H2O.ai was released on 1 Nov 2026. Open weights on Hugging Face.

WAS THIS USEFUL?

ConvAI Innovations Laya

Feedback ⓘ v0.4.21 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's TypeScript client gained a `predictLong` method for scanning long documents.

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

01 **Long-document scanning via predictLong** NEW 55

The TypeScript client adds `predictLong` on the Agent and Router classes, enabling long-document scanning.

– Names exact method and classes but no mechanism detail v0.4.21

WAS THIS USEFUL? 👍 👎

DeepSeek V4 Pro 0813

Feedback ⓘ price cut · Nov 1, 2026

MODEL ↗

DeepSeek V4 Pro 0813 is a chat model from DeepSeek.

Price cut on OpenRouter (deepseek/deepseek-v4-pro-0813): $1.32 → $0.115 in, $3.96 → $4.00 out per million tokens (-91%), first seen 1 Nov 2026.

WAS THIS USEFUL? 👍 👎

Z.ai GLM 4.7

Feedback ⓘ price increase · Nov 1, 2026 MODEL ↗ STABLE

GLM 4.7 is a chat model from Z.ai.

Price increase on OpenRouter (z-ai/glm-4.7): $0.54 → $0.60 in, $1.98 → $2.20 out per million tokens (+11%), first seen 1 Nov 2026.

▾ 1 more release in this issue · 2026-11-01

> retirement scheduled · Nov 1, 2026 MODEL ↗ STABLE
>
> OpenRouter lists a retirement date of 31 Dec 2026 for z-ai/glm-4.7, first seen 1 Nov 2026.

WAS THIS USEFUL? 👍 👎

Z.ai GLM 5.3

Feedback ⓘ price increase · Nov 1, 2026 MODEL ↗

GLM 5.3 is a reasoning model from Z.ai.

Price increase on OpenRouter (z-ai/glm-5.3): $0.0699 → $0.08 in, $1.40 → $3.60 out per million tokens (+157%), first seen 1 Nov 2026.

WAS THIS USEFUL? 👍 👎

Z.ai GLM 5.3 Flash

Feedback ⓘ price cut · Nov 1, 2026 MODEL ↗

GLM 5.3 Flash is a reasoning model from Z.ai.

Price cut on OpenRouter (z-ai/glm-5.3-flash): $0.15 → $0.0285 in, $0.50 → $0.0923 out per million tokens (-82%), first seen 1 Nov 2026.

WAS THIS USEFUL? 👍 👎

◆ Frontier Models

OpenAI gpt-oss-20b

Feedback ⓘ price increase · Nov 1, 2026

MODEL ↗

gpt-oss-20b is a reasoning model from OpenAI.

Price increase on OpenRouter (openai/gpt-oss-20b): $0.018 → $0.03 in, $0.09 → $0.13 out per million tokens (+67%), first seen 1 Nov 2026.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · November 2, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe | Feedback on this issue