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

THE AI TOOLCHAIN

NO. 088

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED NOVEMBER 9, 2026 · EVERY DAY

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

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.

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 47 tools

BUILD

vim ? help

◆ AI Coding Agents

Cotool

Feedback ⓘ changelog · Nov 9, 2026 CHANGELOG ↗

Cotool is an AI security operations platform for investigating alerts, enriching threat intelligence, and automating response workflows.

Cotool adds custom OpenAPI/Swagger integrations, eight new managed log sources, ArmorCode and HaveIBeenPwned integrations, and expanded MCP actions

└──▷ HOW TO FIND IT

Connect a homegrown or third-party API to Cotool by uploading its OpenAPI spec — useful when no built-in integration exists for the tool your team uses.

> 📍 In Cotool, go to Integrations › Add Custom Integration, upload your OpenAPI or Swagger specification file, configure authentication (API key, OAuth, etc.), select the operations to expose, and set approval requirements for sensitive actions.

Ingest 1Password audit events into Cotool so agents can query access and vault activity during an investigation.

> 📍 In Cotool, go to Integrations › Managed Log Ingestion, select `1Password audit events`, supply your 1Password Events API token, and save to begin streaming audit logs.

Let an agent check whether employee email addresses appear in known breach data during an identity-risk triage workflow.

> 📍 In Cotool, go to Integrations › HaveIBeenPwned, enter your HaveIBeenPwned API key, and enable the integration so agents can query breach exposure for any email address during investigation.

Adds custom integrations built from OpenAPI and Swagger specifications, with configurable authentication, operations, and approval requirements, plus new built-in `ArmorCode` integration for security finding management and a `HaveIBeenPwned` integration for breach exposure checks.

Managed log ingestion now supports `AWS`, `GCP`, `Cloudflare`, `CrowdStrike Falcon Data Replicator`, `Tines` audit logs, `GitHub` audit logs, `1Password` audit events, and `SentinelOne` (the latter with threat and alert investigation actions).

Agents now support Google Workspace domain-wide delegation with Alert Center access.

Adds GPT-6.1 Terra to the model catalog.

Cotool MCP now includes Hunt, threat intelligence, detection, ingest health, alert routing, and agent follow-up actions.

+3 more

WAS THIS USEFUL? 👍 👎

yetone magpie

Feedback ⓘ v0.1.3375 CODE ↗ LATEST STABLE

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 adds a Pi packages tab to the library UI for viewing, switching, and copying Pi extensions and packages across local and WSL Pi instances

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.3375
```

git

```
$ git clone --branch v0.1.3375 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3375
```

published under the project's GitHub account · checked 9 Nov Wrong command?

Adds a Pi packages tab (`pi`) to the library panel, visible when a Pi agent is detected, showing extensions and packages for each Pi magpie knows of — the local one and any in WSL distros.

Adds support for reading and writing Pi extensions and packages as Pi's own package manager and `pi config` do, including toggling extensions on and off via the `extensions` array in `settings.json` and copying or switching them between Pi instances.

▾ 12 more releases in this issue · 2026-11-08 → 2026-11-09

v0.1.3374 CODE ↗ STABLE

magpie's gateway now routes bare model names to the asking agent's own subscription first, before falling back to provider order

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.3374
```

```git
$ git clone --branch v0.1.3374 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3374
```

published under the project's GitHub account · checked 9 Nov Wrong command?

v0.1.3373 CODE ↗ STABLE

magpie adds a persistent video generation model setting so the model you pick in Settings › Images › Video generation is used for all video requests

└──▷ GET THIS VERSION ▾

```Docker
$ docker pull ghcr.io/yetone/magpie:0.1.3373
```

```git
$ git clone --branch v0.1.3373 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3373
```

published under the project's GitHub account · checked 9 Nov Wrong command?

Pick a preferred video model in Settings › Images › Video generation — stored as `VideoGen` in settings — and magpie now uses it for all requests that name no model, falling back to the automatic choice when the picked model is no longer available.

Mark any custom provider model as a video model with provider.SetModelKind(ref, provider.KindVideo) so it appears in the video generation model list and can be selected.

New Videomaker() function returns the active video model (as `provider/model`) that requests with no explicit model will use, mirroring the existing image-generation surface.

v0.1.3371 CODE ↗ STABLE

magpie adds a documentation file at `/llms-full.txt` containing every English docs page as Markdown for AI assistants

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.3371
```

git

```
$ git clone --branch v0.1.3371 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3371
```

published under the project's GitHub account · checked 9 Nov Wrong command?

Moves a subscription's warm-up, check-in, and reset controls into the provider's own editor in the GUI, so they are no longer in a separate location.

v0.1.3370 CODE ↗ STABLE

magpie adds automatic System One routing for Volcengine Ark pay-as-you-go providers, skipping Coding and Agent Plan keys

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.3370
```

git

```
$ git clone --branch v0.1.3370 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3370
```

published under the project's GitHub account · checked 9 Nov Wrong command?

Keys on the Ark Coding or Agent Plans are now excluded from System One decision routing, preventing decision requests from being sent to plan endpoints that do not support them.

v0.1.3367 CODE ↗ STABLE

magpie now propagates Remote magpie quota cards through chains of magpies, so a third magpie in a chain sees all accounts behind it

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.3367
```

git

```
$ git clone --branch v0.1.3367 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3367
```

published under the project's GitHub account · checked 9 Nov Wrong command?

Remote magpie quota cards are now propagated down a full chain of magpies: a magpie serving another magpie now passes on the cards it received from its own Remote magpies (`passedOnCards`), so accounts visible to the first magpie in a chain are visible to every downstream magpie, each card tagged with the full path it traveled via the `Via` field.

v0.1.3364 CODE ↗ STABLE

magpie adds routing-group fallback for capped or spent subscription accounts, so requests continue to the next group member

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.3364
```

git

```
$ git clone --branch v0.1.3364 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3364
```

published under the project's GitHub account · checked 9 Nov Wrong command?

v0.1.3362 CODE ↗ STABLE

magpie now records Claude Code compaction calls as `KindCompact` (`compact`) rather than plain turns, closing a gap in usage accounting

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.3362
```

git

```
$ git clone --branch v0.1.3362 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3362
```

published under the project's GitHub account · checked 9 Nov Wrong command?

v0.1.3361 CODE ↗ STABLE

magpie now routes user-marked image models to the images API first, with a chat-completions fallback for providers that lack it

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.3361
```

git

```
$ git clone --branch v0.1.3361 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3361
```

published under the project's GitHub account · checked 9 Nov Wrong command?

User-marked image models (set via `provider.SetModelKind` with `provider.KindImage`) now route requests to the images API (`/v1/images/generations`) first, falling back to chat completions when the provider returns 404 or 405.

v0.1.3360 CODE ↗ STABLE

magpie adds GitHub-linked skill installation so a skill originally copied from a folder can now be updated from its GitHub source

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.3360
```

git

```
$ git clone --branch v0.1.3360 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3360
```

published under the project's GitHub account · checked 9 Nov Wrong command?

Install a skill from its GitHub source to link it to that repository: the library replaces the folder-sourced copy's files and origin in place, exposing it to future checks and updates via the new Links field (`links` in Candidate JSON) and tracking displaced files in `linked` / `kept` state.

v0.1.3358 CODE ↗ STABLE

magpie adds automatic discovery of local provider configurations from Alma, CC Switch, Claude Code and Codex on the Providers screen

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.3358
```

git

```
$ git clone --branch v0.1.3358 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3358
```

published under the project's GitHub account · checked 9 Nov Wrong command?

The Providers screen now discovers importable configurations from Alma, CC Switch, Claude Code and Codex readers automatically, showing a prominent hint when the provider list is empty or a compact import link beside Add provider when providers already exist.

Ignored provider configurations are now remembered across window restarts using opaque fingerprints in local storage, so new or changed configurations can be offered again without reviving dismissed ones.

Discovery now opens the existing picker with only the offered configurations eligible for default selection, while manual import still offers ignored entries.

v0.1.3353 CODE ↗ STABLE

magpie adds a skill router that gives each agent one `magpie-skills` listing instead of a separate link per skill, keeping prompts lean

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.3353
```

git

```
$ git clone --branch v0.1.3353 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3353
```

published under the project's GitHub account · checked 9 Nov Wrong command?

Enable the `SkillRouter` library setting to give an agent a single `magpie-skills` skill in its skills folder — with a `SKILL.md` listing each skill, its description, and a path to its own `SKILL.md` — so the agent loads only the relevant skill's description at task time instead of every skill's.

Skills marked `disable-model-invocation: true` in their `SKILL.md`, skills the user keeps direct via `Skill.Router`, and skills found in `~/.agents/skills` are excluded from the router and keep their own direct links.

The gateway now attaches a skill's body (the `SKILL.md` content sent by Claude Code after a Skill call) to that Skill call's tool result, so the downstream model receives it without the user appearing to have sent it.

v0.1.3348 CODE ↗ STABLE

magpie now shows each agent's icon in the agent filter dropdown on the usage ledger page

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.3348
```

git

```
$ git clone --branch v0.1.3348 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3348
```

published under the project's GitHub account · checked 9 Nov Wrong command?

WAS THIS USEFUL? 👍 👎

Letta Code Feedback ⓘ v0.34.47 NOTES ↗

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

Letta Code now makes the description field optional for Bash and exec_command tools

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/letta-ai/letta-code:0.34.47
```

npm

```
$ npm install -g @letta-ai/letta-code@0.34.47
```

git

```
$ git clone --branch v0.34.47 https://github.com/letta-ai/letta-code.git
# already have the repo? check out this version:
$ git checkout v0.34.47
```

published under the project's GitHub account · signed build provenance · checked 9 Nov

Wrong command?

WAS THIS USEFUL? 👍 👎

corecoder

Feedback ⓘ v0.9.0 NOTES ↗

Corecoder is a minimal AI coding agent (~1,000 lines of Python) inspired by Claude Code. Works with any LLM.

corecoder adds multi-agent team support through the existing `agent` tool, running up to eight named workers with at most four concurrently

└──▷ GET THIS VERSION ▾

pipx

```
$ pipx install corecoder==0.9.0
# already installed? upgrade to this version:
$ pipx install --force corecoder==0.9.0
```

git

```
$ git clone --branch v0.9.0 https://github.com/he-yufeng/CoreCoder.git
# already have the repo? check out this version:
$ git checkout v0.9.0
```

PyPI attestation · checked 9 Nov Wrong command?

Run up to eight named workers as a bounded team through the existing `agent` tool, with at most four running concurrently, isolated histories and checklists, reports in input order, and the parent's consent, hooks, and usage accounting carried through.

corecoder now preserves `reasoning_content` on Kimi and DeepSeek tool-call turns so thinking-protocol context carries into the next request.

corecoder now recognises Kimi's unquoted temperature rejection and falls back to the provider's supported default automatically.

WAS THIS USEFUL? 👍 👎

prflow

Feedback ⓘ v2.64.0 NOTES ↗

prflow is a tool that turns feature requests into review-ready pull requests with implementation, tests, review loop, and docs.

prflow adds branch naming rules, story split tracking, private-repo fact-checking, and library source reading for review checks

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v2.64.0 https://github.com/The01Geek/prflow.git
# already have the repo? check out this version:
$ git checkout v2.64.0
```

└──▷ **USE IT**

Let review checks settle library behavior claims by reading installed source instead of leaving them inconclusive — useful when your codebase depends on third-party libraries whose behavior is questioned during review.

.prflow/config.json

```
prflow_review.dependency_sources:
  - node_modules/some-library/src
  - .venv/lib/python3.11/site-packages/some-other-library
```

Name a feature branch from a naming rule written in your implement extension by configuring it in `/prflow:implement`, supporting conventions like assignee login and ticket ID from the issue body.

List installed-dependency directories in `prflow_review.dependency_sources` so review checklist checks can read library source code instead of leaving behavior claims inconclusive.

`/prflow:spec` now records how a story splits into parts before its first clarification question, keeping an agreed split's remaining parts on the filed issue in landing order.

`/prflow:implement` can now check facts an issue cites from GitHub issues, pull requests, and comments in private repositories.

`/prflow:implement` can now resume a pushed `claude --worktree` branch named for the issue instead of restarting from the base.

+1 more

└──▷ **BREAKING ON UPGRADE**

! Vendored `.prflow/vendor/prflow/` copies must be refreshed to v2.64.0.

Our best guess

Check this claim Claim history

! Delete the `view-head`, `view-base`, `view-head.tmp`, and `view-base.tmp` folders under `.prflow/tmp/review/` to clear review copies left by earlier versions.

Our best guess

Check this claim Claim history

! The implement workflow drops grants for two removed Documentation Needed helpers; shipped workflow configurations change accordingly.

Our best guess

Check this claim Claim history

WAS THIS USEFUL?  

Alibaba Qwen Code

Feedback ⓘ sdk-typescript-v0.1.29

NOTES ↗

Qwen Code is an open-source AI coding agent that lives in your terminal.

Qwen Code adds multi-modal input, session export, concurrent batch runner, LSP support, and a Zed extension across bundled CLI versions

└──▷ GET THIS VERSION ▾

```
$ git clone --branch sdk-typescript-v0.1.29
https://github.com/QwenLM/qwen-code.git
# already have the repo? check out this version:
$ git checkout sdk-typescript-v0.1.29
```

└──▷ USE IT

Cap the context window for a model that reports an incorrect default, preventing truncation errors on long sessions.

```yaml
contextWindowSize: 32768
```

Add `extra_body` support for OpenAI-compatible providers to pass arbitrary request fields.

Resume sessions by passing a session ID with the new `resume` and `continue` options in the TypeScript SDK, with extended `authType` support.

Add `contextWindowSize` configuration key to control the model context window.

Export session history with an `export` command, supporting markdown and HTML output formats.

Add multi-modal input support (image, PDF, audio) across all content generators.

+5 more

└──▷ BREAKING ON UPGRADE

! Negative settings renamed to positive naming: `disable*` settings are now named `enable*` — existing configs using the old names will break.

Our best guess

Check this claim Claim history

WAS THIS USEFUL?  

The Open Engine Zeroshot

Feedback ⓘ v12.0.0 NOTES ↗

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

Zeroshot adds `zeroshot stack` with stacked pull request delivery, target-owned machine sizes, and improved `zeroshot watch` exit status reporting

└──▷ **GET THIS VERSION** ▾

npm

```
$ npm install -g @the-open-engine-company/zeroshot@12.0.0
```

git

```
$ git clone --branch v12.0.0 https://github.com/the-open-engine/zeroshot.git
# already have the repo? check out this version:
$ git checkout v12.0.0
```

signed build provenance · checked 9 Nov Wrong command?

└──▷ **TRY IT**

Validate the linear dependency order of a stack manifest before submission to catch ordering errors early in CI.

```
$ zeroshot stack validate my-stack.stack.json
```

Check which machine sizes a hosted target accepts before authoring a stack manifest that sets a size.

```
$ zeroshot target sizes platform
```

Submit dependent runs as a stack using `zeroshot stack` with manifests declaring `"schema": "zeroshot.stack/v1"` and a required `"delivery"` field set to `merge` or `stacked_pull_requests`.

Use `stacked_pull_requests` delivery mode to open one pull request per run against the previous run's branch without merging, driven by a `pr@2` delivery node in the profile.

Use `merge` delivery mode (requiring a `merge@2` or `merge@3` profile node) to replicate the former merge-plan behaviour, merging each run into the base before dependent runs start.

Query a target's available machine sizes with `zeroshot target sizes NAME`, which outputs a formatted table or JSON of target-owned size identifiers and their descriptions.

`zeroshot watch` now exits 1 and prints a failure summary to stderr when the followed run fails, and exits 0 on success, making it suitable for CI gating.

+2 more

└─▷ **BREAKING ON UPGRADE**

! The shared `RunSize` enum is replaced by an opaque string; omitting size now means the target default. Direct and local targets expose `host` as a size identifier.

Our best guess

Check this claim Claim history

! `zeroshot plan` and manifests using `"schema": "zeroshot.merge-plan/v1"` are no longer accepted; replace them with `zeroshot stack` and `"schema": "zeroshot.stack/v1"`.

Our best guess

Check this claim Claim history

! Protocol types `MergePlan*` are replaced by `Stack*` with no aliases; the Python SDK names `submit_stack`, `StackRequest`, and `StackStatus` replace their `MergePlan` equivalents with no aliases.

Our best guess

Check this claim Claim history

WAS THIS USEFUL?  

coffee-cli

Feedback ⓘ v3.9.4 NOTES ↗ LATEST STABLE

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

coffee-cli adds a Basalt theme and bottom-edge collapse controls for code blocks, diffs, and command output

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v3.9.4 https://github.com/edison7009/Coffee-CLI.git
# already have the repo? check out this version:
$ git checkout v3.9.4
```

coffee-cli adds a Basalt theme — a fine-sand backdrop that becomes the default on first launch alongside the Oat palette and dark mode; the Carbon texture now renders as a quiet fine twill.

coffee-cli now adds an Expand / Collapse tab to the lower-right edge of long code blocks, edit diffs, and command input/output, so readers can collapse content from the bottom after scrolling through it rather than returning to the top.

▾ 1 more release in this issue · 2026-11-08 → 2026-11-09

v3.9.3 NOTES ↗ STABLE

Coffee CLI adds a scheduled Automations feature, per-turn cost and token tracking, and a redesigned Settings page with 24 themes

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v3.9.3 https://github.com/edison7009/Coffee-CLI.git
# already have the repo? check out this version:
$ git checkout v3.9.3
```

Coffee CLI adds an Automations feature that schedules saved prompts to be sent to Claude Code, Codex, DeepSeek Harness, OpenCode, or Pi conversations on a daily, weekday, weekly, interval, or one-time basis, opening the conversation in the foreground when due.

Coffee CLI's DeepSeek Harness now streams replies, displays file edits and commands as cards, and shows running sub-agents and background jobs beside the

composer.

Coffee CLI now shows time elapsed, token count, and cost for every turn in the five desktop conversations, including in history, and displays per-model pricing in the model menu.

Coffee CLI's Settings page now fills the full window, consolidates sounds and shortcuts under General, and includes Coffee Agent configuration, with 24 built-in colour themes (up from the previous set).

Coffee CLI's session list menu now lets you choose which app and icon display style to use, and worktree branches are now named after the project with status shown during creation.

+1 more

WAS THIS USEFUL? 👍 👎

forgedock

Feedback ⓘ v1.53.0 NOTES ↗

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

ForgeDock adds a release job, experiment arm config overrides, conformance-outcome analytics, and account rotation on usage limits

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g forgedock@1.53.0
```

git

```
$ git clone --branch v1.53.0
https://github.com/RapierCraftStudios/ForgeDock.git
# already have the repo? check out this version:
$ git checkout v1.53.0
```

signed build provenance · checked 9 Nov Wrong command?

└──▷ TRY IT

Run a full release from staging to your default branch, resuming if the session is interrupted mid-way.

```
$ forgedock release --resume
```

Adds a `forgedock release` command with durable state, resume, preflight, release PR, promotion, and a post-merge step that confirms artifact publish and opens a version-sync PR.

Adds a release review step that runs the staging panel, diffs on head move, files findings, and opens fix lanes.

Adds a pre-promotion deploy-path rehearsal to the release job to catch broken deploy paths before they reach production.

Adds `issue-watch` release mode that reuses the release job instead of a separate path.

Adds arm-scoped config overrides so a single batch can A/B test lane features such as `explore`, `light`, and `context_pack`.

+3 more

WAS THIS USEFUL?  

juggler Feedback ⓘ v0.8.0 NOTES ↗

The Juggler Code Agent

Juggler adds a one-server-per-machine architecture, shell passthrough via !

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.8.0 https://github.com/juggler-ai/juggler.git
# already have the repo? check out this version:
$ git checkout v0.8.0
```

└──▷ **TRY IT**

Direct a self-hosted or proxy Codex endpoint instead of the default — useful when running Codex behind an internal gateway.

```
$ JUGGLER_CODEX_BASE_URL=https://codex.internal.example.com juggler
```

Use a corporate CA bundle instead of the system certificate store when Juggler runs on a machine with custom TLS inspection.

```
$ SSL_CERT_FILE=/etc/pki/tls/certs/ca-bundle.crt juggler
```

Fold a long conversation down to a summary while keeping the most recent turns intact, then fold everything in one shot.

```
$ /compact
# or, to fold the entire history:
/compact all
```

Point the Codex provider at a custom endpoint with the `JUGGLER_CODEX_BASE_URL` environment variable.

Replace the system certificate store on macOS and Windows by setting `SSL_CERT_FILE` or `SSL_CERT_DIR`.

Run a bash call directly without a model call using `! command` or the `/shell` slash command.

Fold conversation history with `/compact`, which keeps recent turns verbatim; use `/compact all` to fold everything, or run `/compact` inside a sub-thread to compact

only that sub-thread.

Juggler now runs one server per machine that manages multiple sessions, replacing the previous one-server-per-session model.

+5 more

WAS THIS USEFUL? 👍 👎

wrongstack

Feedback ⓘ v1.5.2 NOTES ↗

WrongStack is an AI coding agent that reads code, edits files, runs commands, and reasons through bugs across CLI, TUI, and browser interfaces.

wrongstack now automatically sets WrongMemory as the memory service for profiles that have not chosen one

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.5.2
https://github.com/WrongStack/WrongStack.git
# already have the repo? check out this version:
$ git checkout v1.5.2
```

Installs WrongMemory as the default memory service for any profile that has not yet chosen one, so SAGE memory is active without manual setup.

WAS THIS USEFUL? 👍 👎

keploy

Feedback ⓘ v3.6.225 NOTES ↗

keploy is an open-source platform for creating safe, isolated production sandboxes for API, integration, and E2E testing.

Keploy now replays HTTP mock answers at their originally recorded position and skips background misses from test scoring

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/keploy/keploy:v3.6.225
```

git

```
$ git clone --branch v3.6.225 https://github.com/keploy/keploy.git
# already have the repo? check out this version:
$ git checkout v3.6.225
```

published under the project's GitHub account · checked 9 Nov Wrong command?

Keploy now counts background mock misses during replay against no test, keeping test pass/fail scores accurate when unrelated background traffic goes unmatched.

WAS THIS USEFUL? 👍 👎

mvschwarz OpenRig

Feedback ⓘ v0.7.10 NOTES ↗

OpenRig is a network manager for AI coding agents: build persistent teams with roles, shared context and owned work.

OpenRig adds `rig exec run` for finite Claude Code and Codex queue runs, permission-request visibility in `rig seat status`, and cross-host relay counts in `rig slack status`

└──▷ **GET THIS VERSION** ▾

npm

```
$ npm install -g @openrig/cli@0.7.10
```

git

```
$ git clone --branch v0.7.10 https://github.com/mvschwarz/openrig.git
# already have the repo? check out this version:
$ git checkout v0.7.10
```

named in the project's README · checked 9 Nov Wrong command?

└──▷ **TRY IT**

Run one finite Claude Code turn on a queued task from CI or a script, then check whether it completed.

```
$ rig exec run my-task-row --harness claude --wait
```

Restore login-shell behaviour for Codex seats after upgrading, so rc-defined aliases and shell functions are available to tool commands.

```
$ rig config set launch.codex_login_shell true
```

Adds `rig exec run <row> --harness claude|codex` to run a single finite turn on a queue task, with `--plan`, `--wait`, `status`, and `cancel` subcommands plus `--again` to start a further turn, and cleanly cancels a finite exec via `rig exec cancel` instead of `rig acp cancel`.

View a Claude seat's latest permission request, with suggested rules labelled, using `rig seat status`.

Count cross-host relays currently in flight with `rig slack status`.

Configure Codex tool commands to run in a login shell with `rig config set launch.codex_login_shell true`, deferring to Codex's own `allow_login_shell` setting.

Show spec advisories as launch warnings via `rig up` without blocking launch; unknown `uses` keys now also generate a warning.

+3 more

└──▷ **BREAKING ON UPGRADE**

! The `openrig-core` plugin moves from 0.1.25 to 0.1.26, changing the refocus hook and adding a permission-request hook; replaced files are kept under `$OPENRIG_HOME/backups/managed-refresh/<id>/` — reapply any local edits from there.

Our best guess

Check this claim Claim history

! A single-value `uses.<category>` now selects that exact resource; a member fails to resolve if the pool does not hold it — run `rig agent validate <path>` before upgrading specs.

Our best guess

Check this claim Claim history

! Managed Codex seats now run tool commands in a non-login shell by default, so login- or rc-only setup, shell functions, and aliases may be missing; set `rig config set launch.codex_login_shell true` to restore the previous behaviour.

Our best guess

Check this claim Claim history

! OpenShell users must rebuild the guest image with the release's `guest.py` (see `daemon/assets/openshell/README.md`).

Our best guess

Check this claim Claim history

WAS THIS USEFUL? 👍 👎

rtk

Feedback ⓘ v1.2.0 NOTES ↗

Rtk is a CLI proxy that reduces LLM token consumption by 60-90% on common dev commands. Single Rust binary, zero dependencies

rtk adds directory-folded path listings for `git ls-tree` output

└──▷ GET THIS VERSION ▾

Homebrew

```
$ brew install rtk
# already installed? upgrade to this version:
$ brew upgrade rtk
```

git

```
$ git clone --branch v1.2.0 https://github.com/rtk-ai/rtk.git
# already have the repo? check out this version:
$ git checkout v1.2.0
```

Homebrew lists this version · checked 9 Nov Wrong command?

rtk's `git` filter now folds `ls-tree` path listings by directory instead of expanding every entry on its own line.

WAS THIS USEFUL? 👍 👎

codealta

Feedback ⓘ 4.1.0 NOTES ↗

Codealta is an agentic Development Environment

CodeAlta adds a Fleet console, always-on node boot, side questions, page annotations, and chained automations in 4.1.0

└──▷ GET THIS VERSION ▾

```
$ git clone --branch 4.1.0 https://github.com/CodeAlta/CodeAlta.git
# already have the repo? check out this version:
$ git checkout 4.1.0
```

.
.
.

└──▷ TRY IT

Persist a fleet node across reboots on a build server so sessions and automations survive restarts without manual intervention.

```
$ altanode install --system
```

Ask the active model a quick clarifying question mid-session without polluting the conversation history.

```
$ /ask What files does this module expose as its public API?
```

Install a node or server as a system service with `altanode install --system` or `altaserver install --system`, and tick Always on when adding a machine over SSH, so the node starts at boot as a Windows service, a LaunchDaemon on macOS, or a `systemd` unit on Linux.

Ask the model of a live session a side question without adding it to the conversation by typing `/ask` or clicking the speech-bubble icon; all providers answer, including Claude Code, Codex CLI and ACP agents, and earlier answers stay in a list.

Open `Settings > Fleet` to reach a new Fleet console with an Overview showing machines online, server status, and what still needs attention, plus sortable, filterable, paginated tables for machines, people, devices, policies and the audit log.

Sign a machine in to Codex, GitHub Copilot, Claude Code, Mistral Vibe, ACP agents, MCP servers and `gh` from the desktop without a terminal or SSH, using the new Sign in… action on the machine's page.

Annotate project pages in the CodeAlta browser with Annotate, mark elements, attach a mockup, and write what should change; the notes reach the session together with the source files where each element is defined.

+5 more

WAS THIS USEFUL? 👍 👎

◆ Coding Agent Add-ons

Diagram Design

Feedback ⓘ changes since 2026-10-14 CODE ↗

Diagram Design is a Claude Code skill that generates editorial HTML and SVG diagrams and imports draw.io or Mermaid sources.

Diagram Design: Adds native Qoder plugin support.

.

Adds a box plot variant to the scatter chart type for rendering five-number summaries.

WAS THIS USEFUL? 👍 👎

cezar Feedback ⓘ v0.19.0 NOTES ↗

cezar is an open-source orchestrator and ADE for running Claude Code, Codex, OpenCode, Pi and other AI coding agents in parallel — locally or 24/7 on your own server.

cezar adds Gemini CLI, Kimi Code, Antigravity and omp agent runners, HTML artifacts, Assess Impact, fallback providers, and export/import for automations and config

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g @open-mercato/cezar@0.19.0
```

git

```
$ git clone --branch v0.19.0 https://github.com/open-mercato/cezar.git
# already have the repo? check out this version:
$ git checkout v0.19.0
```

signed build provenance · checked 9 Nov Wrong command?

Adds `cez handoff log|resume` to run agents directly from the handoff file without custom scripts, and introduces four new agent runners: Gemini CLI over ACP, Kimi Code over ACP, Antigravity CLI (`agy`), and `omp` (pi's successor).

Agents now publish HTML artifacts that show live inside the task; Assess Impact, the first feature built on them, analyzes PRs and specs.

`cezar serve` now joins a verified running cockpit and opens the current project there instead of starting a duplicate server.

Adds live Claude usage gauges (5-hour and 7-day windows plus context fill) in the task header, toggled via Settings → Appearance, and an Effort pill next to Model for picking reasoning effort per task for Codex and Claude, with levels discovered per model.

Adds Run First to move a queued task to the front of its project's queue, and an option to remove a finished task's worktree from its menu while keeping the branch, task, and transcript.

+5 more

└──▷ BREAKING ON UPGRADE

! Jira and Linear tokens move from plaintext files to encrypted vault records; older cezar binaries cannot read migrated records — stop older processes before upgrading, and use a separate home directory to downgrade.

Our best guess

Check this claim Claim history

! Tracker automation agents no longer receive Jira/Linear tokens; they now receive server-fetched issue context and the `cez tracker read|comment|transition` commands, scoped to their task's project and issue. Comments and status changes require an explicit grant in the automation.

Our best guess

Check this claim Claim history

WAS THIS USEFUL? 👍 👎

◆ Model Training & Fine-Tuning

fizgig

Feedback ⓘ v9.2.5 NOTES ↗ LATEST STABLE

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

Fizgig adds a MOSS-Image Turbo Train preset with per-image adaptive learning rate and reverses MOSS-Image's Clean-end share defaults

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/shootthesound/fizgig:9.2.5
```

git

```
$ git clone --branch v9.2.5
https://github.com/shootthesound/Fizgig.git
# already have the repo? check out this version:
$ git checkout v9.2.5
```

published under the project's GitHub account · checked 9 Nov Wrong command?

New MOSS-Image Turbo Train (Experimental) preset for MOSS-Image uses rank 16, 0.25 MP, Muon, Adaptive LR between 2e-4 and 4e-4, per-image tracking, and per-image adaptive LR, so each image gets its own learning rate pace during training.

MOSS-Image's Clean-end share order is now `30:7:60:8:3` for LoRA, Style, and Slider presets, spending 30% of steps on nearly clean images until epoch 7, then 60% from epoch 8, with every 3rd epoch after that at 30% and half the learning rate; Fine-tune keeps `11:4:30:8:2`.

Faster, smaller previews is now on by default on cards under 20 GB, with the option to untick it for full-size previews.

Per-image loss view adds a Refresh button beside its sort menus, and changing either menu now loads the latest results automatically.

▾ 2 more releases in this issue · 2026-11-08

> v9.2.4 NOTES ↗ STABLE
>
> Fizgig adds multi-line edit instructions with per-line previews and an optional 768 px faster-preview mode for MOSS-Image cards under 20 GB

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/shootthesound/fizgig:9.2.4
```

git

```
$ git clone --branch v9.2.4
https://github.com/shootthesound/Fizgig.git
# already have the repo? check out this version:
$ git checkout v9.2.4
```

published under the project's GitHub account · checked 9 Nov Wrong command?

The Samples tab now has a Faster, smaller previews (768 px) tick for MOSS-Image cards under 20 GB, off by default, so you can trade preview quality for speed during long runs; the gallery marks previews made at the smaller size.

The Edit instruction box on the Edit pairs card now accepts one instruction per line, and every line is previewed on your test photo each round, so a LoRA trained on a family of edits shows all of them at once.

Training speed reported on the run card now excludes everything between epochs (saves, checks, and previews), reflecting the speed of training steps only.

v9.2.2 NOTES ↗ STABLE

Fizgig adds faster MOSS-Image training previews via a Turbo LoRA tick on the Samples tab, and pod compile caches that survive restarts

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/shootthesound/fizgig:9.2.2
```

git

```
$ git clone --branch v9.2.2
https://github.com/shootthesound/Fizgig.git
# already have the repo? check out this version:
$ git checkout v9.2.2
```

published under the project's GitHub account · checked 9 Nov Wrong command?

> Enable or disable Turbo LoRA previews for MOSS-Image training with the 'Use the MOSS-Image Turbo LoRA for previews' tick on the Samples tab, with a Strength control beside it — Studio, LoRA Royale and the Profiler all follow the same setting.
>
> Pods now keep compiled blocks on the pod's volume so a Resume after a restart, Pause or Move reuses them instead of recompiling from scratch.

WAS THIS USEFUL? 👍 👎

◆ Speech & Voice AI

Parakeet ASR Server

Feedback ⓘ v0.11.0 NOTES ↗

Parakeet ASR Server transcribes audio on CPUs or GPUs using NVIDIA Parakeet ONNX models through a Whisper-compatible API.

Parakeet ASR Server now regenerates models from the official NVIDIA checkpoint and adds local attention encoding for long audio, cutting WER from 2.9% to 1.9%.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.11.0
https://github.com/achetronic/parakeet.git
# already have the repo? check out this version:
$ git checkout v0.11.0
```

Parakeet ASR Server now generates its own ASR models directly from the official NVIDIA checkpoint, removing the dependency on third-party ONNX conversion.

The encoder now runs with local attention (`rel_pos_local_attn`), reducing word error rate on long audio from 2.9% to 1.9% on a 29-minute reference.

WAS THIS USEFUL? 👍 👎

AGENT

◆ MCP Servers & Tooling

funplay-cocos-mcp

Feedback ⓘ v0.7.1 NOTES ↗

Funplay-cocos-mcp is the Most Advanced MCP Server for Cocos Creator Editor with `execute_javascript`, prompts/resources, screenshots, asset dependency validation, tool profiles, one-click client configuration, and editor automation.

funplay-cocos-mcp adds a bilingual Contact & Feedback panel with author email and QQ group copy buttons

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g funplay-cocos-mcp@0.7.1
```

git

```
$ git clone --branch v0.7.1 https://github.com/FunplayAI/funplay-cocos-mcp.git
# already have the repo? check out this version:
$ git checkout v0.7.1
```

signed build provenance · checked 9 Nov Wrong command?

└──▷ HOW TO FIND IT

Reach the author or community directly from inside Cocos Creator without leaving the editor.

📍 In the Cocos Creator menu bar, go to Funplay › Contact & Feedback, then use the copy buttons to copy the author's email or QQ group number 1108522331.

WAS THIS USEFUL? 👍 👎

godot-mcp

Feedback ⓘ v0.37.0 NOTES ↗

Godot-MCP — Model Context Protocol (MCP) integration for the Godot Engine. dev.

godot-mcp adds Install All and Update All dock actions plus a `update` CLI command that refreshes the addon, extensions, and CLI itself in one step

└──▷ **GET THIS VERSION** ▾

npm

```
$ npm install -g godot-cli@0.37.0
```

git

```
$ git clone --branch v0.37.0 https://github.com/IvanMurzak/Godot-MCP.git
# already have the repo? check out this version:
$ git checkout v0.37.0
```

signed build provenance · checked 9 Nov Wrong command?

└──▷ **TRY IT**

Keep a project's addon, extensions, and the CLI binary current without running multiple commands.

```
$ godot-cli update ./MyGodotProject
```

Update the project addon, all extensions, and the CLI itself in one step with the `update` CLI command.

Install or update all extensions at once from the Godot Editor dock using the new Install All and Update All buttons.

WAS THIS USEFUL? 👍 👎

chap

Feedback ⓘ v0.3.5 NOTES ↗

CHAP, the Collaborative Human Agent Protocol, is a MCP/A2A-compatible runtime for auditable human-agent work: approvals, overrides, handoffs, escalation and verifiable evidence logs.

chap adds `@brightbeamai/chap-server`, `@brightbeamai/chap-code-gate`, Python CLI servers, `review.acknowledge`, WebAuthn signatures, MCP 2026-07-28, A2A 1.0, and framework b

└──▷ **GET THIS VERSION** ▾

pip

```
$ pip install chap-coordinator==0.3.5
# already installed? upgrade to this version:
$ pip install --upgrade chap-coordinator==0.3.5
```

npm

```
$ npm install @brightbeamai/chap-coordinator@0.3.5
```

git

```
$ git clone --branch v0.3.5 https://github.com/BrightbeamAI/chap.git
# already have the repo? check out this version:
$ git checkout v0.3.5
```

named in the project's README · checked 9 Nov Wrong command?

└──▷ **TRY IT**

Start the CHAP desk server from Python to serve the review UI, MCP, and read API as an ASGI app behind your existing web framework.

```
$ chap-coordinator serve
```

Bring an existing generated project up to 0.3.5, updating the bundled server and code gate copies and adding any new required configuration.

```
$ npx create-chap-app@0.3.5 upgrade my-project
```

Create a workspace that requires all reviewers to approve before a task completes, so no single reviewer can unilaterally decide.

```json
{
  "jsonrpc": "2.0",
  "id": 1,
  "method": "workspace.create",
  "params": {
    "name": "my-workspace",
    "review_rule": "all_approve"
  }
}
```

Adds the CHAP desk server as installable packages: run it from Python with `chap-coordinator serve`, `chap-coordinator mcp`, and `chap-coordinator a2a` commands, with `chap_coordinator.client` exposing `HttpCoordinator`, `Participant`, and `Signer`, and `chap_coordinator.server` exposing the desk as an ASGI app; or install it for Node http, Express, or Fastify as `@brightbeamai/chap-server`, mounted at any prefix, whose client's `signedRead` signs a read with a member's key, with the desk signing via passkey where the browser has one.

Bring a generated project forward or add CHAP to an existing project with `create-chap-app upgrade` and `create-chap-app add`; `--adopt-settings` takes the template's settings and moves the store aside where they change signing or the chain.

Improves protocol support: `chapMcpHttpHandler` serves MCP over HTTP answering protocol version 2026-07-28 alongside the handshake versions, with `chap-mcp-server --url` forwarding to a desk, and `@brightbeamai/chap-coordinator-a2a` adds A2A 1.0 support via `@a2a-js/sdk` 1.3 while continuing to answer 0.3 clients, with the Python `a2a` extra installing `a2a-sdk` with its HTTP server.

Add a `review.acknowledge` call so reviewers can record that they have opened a review, separating the time a review waited unopened from the time it was read.

Verify WebAuthn passkey assertions with a `webauthn:` tag, checked against the origins listed on the key.

+3 more

└─▷ **BREAKING ON UPGRADE**

! Node.js 20 is no longer supported; Node 22 or later is now required.

Our best guess

Check this claim Claim history

! Many calls that 0.3.4 accepted are now refused with `-32011`: `review.request` before a review is open; `decide.*`, `abstain.declare`, or `review.acknowledge` from the work's authors; `role: "admin"` at `participant.join` except from the first member or a named admin invitation; and joins under a member's name with keys that member does not hold.

Our best guess

Check this claim Claim history

! A signed request identical to one already accepted is now refused with `-32075` and `data.accepted_at_seq` — replaying an accepted envelope no longer succeeds.

Our best guess

Check this claim Claim history

! A workspace that advertises `security-signed/1.0` or an identity profile cannot drop it with `workspace.set_profiles`.

Our best guess

Check this claim Claim history

! A paused participant can no longer lift its own pause or leave and rejoin to come back unpaused.

Our best guess

Check this claim Claim history

! `__proto__`, `constructor`, and `prototype` are refused as idempotency keys.

Our best guess

Check this claim Claim history

! The desk server and reference servers now check `Host` and `Origin`, accept JSON only, and send no `Access-Control-Allow-Origin`; a desk server admits only the members `chap.config.json` names.

Our best guess

Check this claim Claim history

! An Ed25519 signature in anything other than canonical standard base64 is refused.

Our best guess

Check this claim Claim history

! The MCP and A2A libraries now refuse a person as the participant they act as.

Our best guess

Check this claim Claim history

! The default `require_signatures` in a new project is now `"humans"` (agents work unsigned, people sign); the production template keeps `true`.

Our best guess

Check this claim Claim history

! The `mcp-gate` template and the Python desk server now listen on port 8792 instead of their previous port.

Our best guess

Check this claim Claim history

WAS THIS USEFUL?  

homelab-monitor

Feedback ⓘ v0.38.0 NOTES ↗

Homelab-monitor is a self-hosted homelab dashboard in one container — GPU, local-AI VRAM, Docker, systemd, energy costs. Read-only MCP server so AI agents can explore it.

homelab-monitor adds an optional AI agent that can read, propose, and act on your lab — plus dashboard login with API key scopes

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.38.0
https://github.com/SikamikanikoBG/homelab-monitor.git
# already have the repo? check out this version:
$ git checkout v0.38.0
```

└──▷ TRY IT

Turn on the AI agent for a homelab that already runs Ollama, keeping every host in read-only Passive mode until you decide to promote them.

```
$ AGENT_ENABLED=1 ADMIN_PASSWORD=changeme docker run -d --name
homelab-monitor sikamikaniko123/homelab-monitor:0.38.0
```

Re-enable the host run endpoint for a migration script that depends on it, so calls to `/api/hosts/<name>/run` are accepted and logged.

```
$ ENABLE_HOST_RUN=1 docker run -d --name homelab-monitor
sikamikaniko123/homelab-monitor:0.38.0
```

Enable an AI agent for your lab with `AGENT_ENABLED=1`; the agent uses read tools from the MCP server and surfaces answers via a new Agent tab and a floating Ask button on every page.

Connect the agent to models already on your network — Ollama, vLLM, llama.cpp, LM Studio, LocalAI — or supply your own key for OpenAI, Anthropic, Gemini (experimental), Azure OpenAI, Groq, Mistral, DeepSeek, xAI, OpenRouter, or Together; keys are write-only and never leave the hub.

Run ten built-in automation templates (Docker cleanup, model housekeeping, disk-space watch, system maintenance, data & log cleanup, GPU health, update report, cost report, container health, backup freshness) or a custom prompt on a schedule in

Report, Propose, or Act-within-policy mode, with alerts to configured channels and proposals queued under Approvals.

Review every agent run step by step in the Activity view — what it read, what policy decided, who approved — and export the log as JSON or CSV; a copy of every action also appears in `docker logs`.

Add an optional dashboard login by setting a password via Settings → Security or the `ADMIN_PASSWORD` environment variable at boot; without one, the dashboard works exactly as before.

+3 more

└─▷ **BREAKING ON UPGRADE**

! The `/api/hosts/<name>/run` endpoint is now disabled by default; any script that calls it will get no response until `ENABLE_HOST_RUN=1` is set.

Our best guess

Check this claim Claim history

! Database backups no longer contain secrets — webhook URLs, tokens, passwords, AI provider keys, the password hash, and all sessions are blanked in the download; after a restore, all credentials must be re-entered.

Our best guess

Check this claim Claim history

WAS THIS USEFUL? 👍 👎

memwhale

Feedback ⓘ v0.26.0 NOTES ↗

Memwhale is a 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 adds honest recall scoring and a semantic search floor, with saved lessons now leading context and search results

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.26.0 https://github.com/wuisabel-gif/MemWhale.git
# already have the repo? check out this version:
$ git checkout v0.26.0
```

└──▷ TRY IT

After saving a lesson for a build failure, confirm it surfaces at the top of context before handing it to an agent or chat.

```
$ mw remember "linker cc not found -> run: xcode-select --install"
mw context --last-error
```

Saves lessons via `mw remember` now appear first in `mw context --last-error` output, ranked at importance 0.8 above a failed command's 0.65, and lead search results over raw failures.

Semantic search now abstains when cosine similarity falls below 0.5 (measured with `nomic-embed-text`), so search can return `Insufficient evidence` instead of always producing a low-confidence result.

`memorywhale-core` 0.13.0 adds `SEMANTIC_MATCH_FLOOR` and `BuiltinEngine` now overrides `retrieve_matches` to enforce the floor.

`memorywhale-cli` adds `follows_fix` and the `mw_follows` SQL function so that recall scoring in `mw doctor`, `mw recall`, and `/recall` counts a note as followed only when its actual fix command ran.

└──▷ BREAKING ON UPGRADE

! Scripts that read the first result of `mw search` may now see a saved note where they previously saw a command, because saved notes rank higher after this release.

Our best guess

Check this claim Claim history

! The recall score shown by `mw doctor` may drop after upgrading, because the old score counted unrelated commands as 'followed'.

Our best guess

Check this claim Claim history

WAS THIS USEFUL? 👍 👎

pullmd

Feedback ⓘ v3.18.2 NOTES ↗

Pullmd is a self-hosted URL- and file-to-Markdown service for humans and AI agents - web pages, documents, images, audio, YouTube. PWA + REST + MCP + Claude Code skill, Reddit-aware, refreshable share links.

pullmd adds automatic article-series joining for pages that split content across multiple `<article>` blocks

└──▷ **GET THIS VERSION** ▾

Docker

```
$ docker pull ghcr.io/aeternalabshq/pullmd:3.18.2
```

git

```
$ git clone --branch v3.18.2
https://github.com/AeternaLabsHQ/pullmd.git
# already have the repo? check out this version:
$ git checkout v3.18.2
```

published under the project's GitHub account · checked 9 Nov Wrong command?

└──▷ **USE IT**

Opt out of article-series joining when you need each `<article>` block fetched as a separate document.

.env

```
PULLMD_ARTICLE_SERIES=off
```

Pages that split a single piece of writing across several `<article>` blocks are now joined into one output; control this behaviour with the `PULLMD_ARTICLE_SERIES=off` environment variable.

WAS THIS USEFUL? 👍 👎

◆ AI Agent Frameworks

magic-context

Feedback ⓘ v0.49.44 NOTES ↗

magic-context is a memory system for AI agents that provides unbounded context and manages itself across sessions.

magic-context now estimates token usage for providers that report none, so context pressure handling works without cache expiry as a trigger

└──▷ GET THIS VERSION ▾

npm

```
$ npm install @cortexkit/magic-context@0.49.44
```

git

```
$ git clone --branch v0.49.44 https://github.com/cortexkit/magic-context.git
# already have the repo? check out this version:
$ git checkout v0.49.44
```

signed build provenance · checked 9 Nov Wrong command?

Providers that report no token usage now get context pressure handling: magic-context estimates the size of the sent request and uses that estimate to decide when the execute threshold (65% by default) is reached, counting as force-band pressure only at 95% or more.

An expired provider `cache_ttl` no longer triggers a prompt rebuild; rebuilds now happen only at the execute threshold or on a model or system prompt change, keeping `cache_ttl` for diagnostics only.

WAS THIS USEFUL? 👍 👎

agentlas-os

Feedback ⓘ v1.2.91 NOTES ↗

Agentlas-os is an agent OS: keep specialist agents in a hub, spin up a temporary orchestrator per task. Local-first, works with any model.

Agentlas OS adds cost-ceiling fallback so stages now run on the host model when a worker tier cannot be verified, instead of ending unresolved

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.2.91 https://github.com/agentlas-ai/Agentlas-OS.git
# already have the repo? check out this version:
$ git checkout v1.2.91
```

Cost ceilings set via `/agentlas-orch` (for example `worker=economy`) no longer halt work: when the host did not report a price tier or the model is above the ceiling, the allocation receipt now records `cost_tier_unverified` or `current_model_exceeds_cost_policy` and the stage continues on the current host model instead of ending as unresolved.

Parent requests the host cannot satisfy — an unreported tier or effort level — now fall back to the host model the same way, while operator pins, provider pins, and exact model requests are still never replaced silently.

WAS THIS USEFUL? 👍 👎

Graphicle AI Graphicle

Feedback ⓘ v0.1.14 NOTES ↗

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

Graphicle adds optional PostgreSQL storage, a `graphicle import-sqlite` migration command, and a WAL-capping checkpoint for SQLite under constant reads

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.14 https://github.com/Graphicle-AI/graphicle.git
# already have the repo? check out this version:
$ git checkout v0.1.14
```

Adds a PostgreSQL store: set `GRAPHICLE_DATABASE_URL` to store data in PostgreSQL 16 or newer, with each data folder getting its own schema named in the folder's `postgres.json`; migrate an existing SQLite data folder once with `graphicle import-sqlite` after setting `GRAPHICLE_DATABASE_URL` (the SQLite file is left untouched); override the per-session statement timeout (default 600 seconds) with `GRAPHICLE_DATABASE_STATEMENT_TIMEOUT`; and allow plaintext connections to a remote PostgreSQL server, normally refused, by setting `GRAPHICLE_DATABASE_ALLOW_PLAINTEXT=1`.

Track slow write-lock waits in a project's status output via the new `slow_waits`, `longest_wait_seconds`, and `longest_wait_at` fields under `write_lock`.

SQLite's write-ahead log now caps at 64 MiB and is checkpointed and truncated, preventing unbounded growth under constant reads (previously observed at 8.3 GB on one service).

Reduces write-lock contention and improves its reporting: MCP messages and the `graph_as_of`, episode, and cross-project edge tools no longer take the write lock to check project existence, instead reading a snapshot, and budget checks for requests waiting on the write lock are now recorded as `unrecorded_requests` in the usage report when counts are held in memory.

Adds `bench/store-load/`, a load harness that indexes a large repository into a store while 10 to 33 MCP clients read concurrently, recording read latency, failures, memory, WAL size, and fsyncs.

└──▷ BREAKING ON UPGRADE

! When `GRAPHICLE_DATABASE_URL` is set, any data folder that has only `reviews.sqlite3` is refused by every command — including read-only ones such as

`graphicle graph` — until `graphicle import-sqlite` has been run on it.

Our best guess

Check this claim Claim history

WAS THIS USEFUL? 👍 👎

CrewAI

Feedback ⓘ 1.15.38 NOTES ↗

CrewAI is an open-source Python framework for building multi-agent workflows using role-based AI agent Crews and event-driven Flows.

CrewAI adds `--no-open` to `crewai eval` for headless use, sets gpt-6-luna as the new default model, and now reads spreadsheet URLs row-by-row in agent inputs

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch 1.15.38 https://github.com/crewAIInc/crewAI.git
# already have the repo? check out this version:
$ git checkout 1.15.38
```

└──▷ **TRY IT**

Run an evaluation in a headless CI environment where no browser should be opened.

```
$ crewai eval --no-open
```

Run `crewai eval` without opening a browser by passing `--no-open`, enabling headless CI and automated evaluation pipelines.

Tag evaluation tracing spans with a target and a per-evaluation ID, making it easier to correlate traces across runs in observability tooling.

CrewAI now uses `gpt-6-luna` as the default model everywhere, replacing the previous default.

Agents can now read spreadsheet URLs in their input and apply tools over every row, enabling row-by-row spreadsheet processing without custom glue code.

└──▷ **BREAKING ON UPGRADE**

! The default model is now `gpt-6-luna` everywhere; any project relying on the previous implicit default will switch models on upgrade.

Our best guess

Check this claim Claim history

WAS THIS USEFUL? 👍 👎

pi-fabric

Feedback ⓘ v0.134.5 NOTES ↗

Pi-fabric is a programmable tool and agent runtime for Pi

pi-fabric adds bounded diagnostics via `PI_FABRIC_DEBUG_TOOL_CAPTURE=1` and expands managed-install coverage to nested writes, edits, bash, and blocking `tool_call` hooks

└──▷ **GET THIS VERSION** ▾

npm

```
$ npm install pi-fabric@0.134.5
```

git

```
$ git clone --branch v0.134.5 https://github.com/fabric-runtime/pi-fabric.git
# already have the repo? check out this version:
$ git checkout v0.134.5
```

signed build provenance · checked 9 Nov Wrong command?

└──▷ **TRY IT**

Troubleshoot why a nested tool hook is not being captured without exposing sensitive arguments or paths in logs.

```
$ PI_FABRIC_DEBUG_TOOL_CAPTURE=1 bun run your-fabric-entry.ts
```

Enable bounded capture diagnostics — showing hub installation and anchor matches without logging arguments, commands, paths, or results — by setting `PI_FABRIC_DEBUG_TOOL_CAPTURE=1`.

Managed-install coverage now extends to nested writes, edits, bash, captured extension tools, and blocking `tool_call` hooks, and supports an explicit release-binary integration probe.

WAS THIS USEFUL? 👍 👎

persona

Feedback ⓘ @runtypelabs/persona@4.37.1 NOTES ↗

Persona is a create agentic front-end experiences for the web in VanillaJS. Lightweight, extensible, and WebMCP-native.

Persona adds compact reconnect notices, terminal execution failure display, and an `executionErrorDisplay` setting for host-controlled error rendering

└──▷ GET THIS VERSION ▾

npm

```
$ npm install @runtypelabs/persona@4.37.1
```

git

```
$ git clone --branch @runtypelabs/persona@4.37.1
https://github.com/runtypelabs/persona.git
# already have the repo? check out this version:
$ git checkout @runtypelabs/persona@4.37.1
```

signed build provenance · checked 9 Nov Wrong command?

Hosts that render their own failure notice can suppress duplicate transcript messages by setting `executionErrorDisplay: "host"`.

Persona now shows compact connection notices with a Reconnect action when a streaming response is interrupted, and preserves the execution checkpoint so Reconnect resumes the same response.

Persona now shows terminal execution failures in the transcript while keeping the connection indicator online.

Persona now honors `features.artifacts.allowedTypes` for streamed artifacts, including transcript blocks and continuation events.

WAS THIS USEFUL? 👍 👎

◆ Agent Sandboxes

containarium

Feedback ⓘ v0.132.0 NOTES ↗

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

Containarium adds opt-in label-driven connection dispatch and deployer-supplied extra lock support in v0.132.0

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/footprintai/containarium:v0.132.0
```

pipx

```
$ pipx install containarium-telemetry==0.132.0
# already installed? upgrade to this version:
$ pipx install --force containarium-telemetry==0.132.0
```

git

```
$ git clone --branch v0.132.0
https://github.com/FootprintAI/Containarium.git
# already have the repo? check out this version:
$ git checkout v0.132.0
```

published under the project's GitHub account · PyPI attestation · checked 9 Nov Wrong command?

Make label-driven connection dispatch opt-in per connection with the change from #3569/slice 1 (#3572), so individual connections can selectively enable the label dispatch path.

Accept a deployer-supplied extra lock (#3570/#3571), letting deployers inject an additional lock alongside the daemon's own locking.

WAS THIS USEFUL? 👍 👎

◆ Applied AI Agents

Feynman

Feedback ⓘ v0.5.66 NOTES ↗

Feynman is a personal AI research assistant that searches papers and the web, builds literature reviews, and produces cited research reports.

Feynman adds controlled subagent drain-stop, automatic artifact cleanup, and strict provider routing for pi-subagents 0.87.0

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g @companion-ai/feynman@0.5.66
```

git

```
$ git clone --branch v0.5.66 https://github.com/Companion-
Inc/feynman.git
# already have the repo? check out this version:
$ git checkout v0.5.66
```

signed build provenance · checked 9 Nov Wrong command?

└──▷ USE IT

Set how many days to keep subagent artifacts in nested session folders before automatic cleanup removes them.

yaml

```
artifactConfig:
  cleanupDays: 7
```

Clean up old subagent artifacts in nested session folders automatically after the period set in `artifactConfig.cleanupDays`.

Feynman now enforces strict provider routing for subagents written as `provider/id`, failing at launch with a named error instead of silently billing an alternative provider's account.

A Pi install missing `@earendil-works/pi-schema` now fails a background launch immediately with an error naming the missing package, rather than silently misbehaving.

└──▷ BREAKING ON UPGRADE

! Subagents written as `provider/id` no longer fall back to another provider's copy of the model — the launch now fails with an error naming the alternative, so any workflow relying on silent fallback will stop working.

Our best guess

Check this claim Claim history

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Inference Providers & GPU Clouds

AWS Amazon Bedrock

Feedback ⓘ changelog · Nov 9, 2026

CHANGELOG ↗

Amazon Bedrock provides managed foundation-model inference, customization, knowledge bases, and APIs for generative AI applications.

Amazon Bedrock adds watermarking documentation covering which models embed watermarks in output and how to detect them.

NEW SUPPORTED PLATFORM Kimi K3.1 Pro - Highspeed model support

WAS THIS USEFUL? 👍 👎

Cloudflare Workers AI

Feedback ⓘ changelog · Nov 9, 2026

CHANGELOG ↗

Cloudflare Workers AI runs open-weight models, including the Clef decision models, as serverless inference on Cloudflare's global network.

Cloudflare Workers AI adds model aliases for pool routing in AI Gateway, automatic Durable Object call retries, passkey login for Access, and fast flux UEBA detection in Clou

└──▷ **USE IT**

Tune the runtime retry window for a specific Durable Object stub so session-sensitive calls do not get retried too aggressively.

javascript

```
const stub = env.CART.getByName('user-123', { retry: { maxRetries: 2, timeoutMs: 5000 } });
```

Block keyboard data entry in an isolated browser session while keeping find-in-page, copy, and zoom available to users.

```
$ curl -X PUT https://api.cloudflare.com/client/v4/accounts/{account_id}/gateway/rules/{rule_id} \
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{"action": {"type": "isolate", "value": {"biso_admin_controls": {"version": "v2", "keyboard": "enable_read_only_actions"}}}}'
```

Route AI requests across a pool of models using model aliases in AI Gateway by calling an alias name as the `model` field on `POST /accounts/{account_id}/ai/run`, with the `cf-aig-gateway-id` header selecting the gateway; AI Gateway tries models in order or uses smart routing, letting you swap models without redeploying.

Adds new Durable Object retry controls: automatic retries for fetch() and RPC calls to a Durable Object are now built into the runtime (up to four retries within 10 seconds) for retryable errors occurring before reaching the Durable Object, requiring Wrangler 4.158.0 or later; you can mark a method as safely retryable by importing `{ retryable }` from `cloudflare:durable-objects` and registering it with retryable(prototype.methodName) (methods not marked are never run twice); and you can control retries per stub by passing `retry: { maxRetries: <0–10>, timeoutMs:`

`<500-60000> }` to getByName() or get(), with `maxRetries: 0` disabling runtime retries where your code handles them.

Block data entry in Browser Isolation while preserving find-in-page, copy, and zoom by setting Keyboard to Allow read-only actions in an isolation policy, or via the API by setting `keyboard` to `enable_read_only_actions` inside the `biso_admin_controls` object with `version` set to `v2`.

Review provider key activity in AI Gateway under Provider Keys, where keys are grouped by provider with masked previews, aliases, and last-used and last-updated dates; the Key security section flags keys with activity older than 90 days for rotation or removal.

Enable fast flux detection as a predefined UEBA risk behavior in Cloudflare One by navigating to Risk score > Risk behaviors in the Cloudflare Zero Trust dashboard (Enterprise plans only).

+3 more

└─▷ **BREAKING ON UPGRADE**

! Cloudflare Workers AI models `@cf/deepseek-ai/deepseek-r1-distill-qwen-32b`, `@cf/nvidia/nemotron-3-120b-a12b`, `@cf/deepseek-ai/deepseek-v4-pro-0813`, `@cf/deepseek-ai/deepseek-v4-flash-0731`, `@cf/qwen/qwq-32b`, `@cf/qwen/qwen3-30b-a3b-fp8`, `@cf/moonshotai/kimi-k2.6`, `@cf/moonshotai/kimi-k2.7-code`, `@cf/zai-org/glm-5.2`, `@cf/zai-org/glm-5.3-flash`, `@cf/meta-llama/llama-2-7b-chat-hf-lora`, `@cf/google/gemma-2b-it-lora`, `@cf/google/gemma-7b-it-lora`, `@cf/meta/llama-3.1-8b-instruct-fp8`, `@cf/meta/llama-3.2-1b-instruct`, `@cf/meta/llama-3.2-3b-instruct`, `@cf/myshell-ai/melotts`, `@cf/stabilityai/stable-diffusion-xl-base-1.0`, `@cf/runwayml/stable-diffusion-v1-5-inpainting`, `@cf/bytedance/stable-diffusion-xl-lightning`, `@cf/lykon/dreamshaper-8-lcm`, `@cf/openai/whisper-tiny-en`, `@cf/llava-hf/llava-1.5-7b-hf`, and `@cf/google/embeddinggemma-300m` stop returning responses on November 20, 2026; requests will fail with HTTP `410 Gone` and error code 8000.

Our best guess

Check this claim Claim history

WAS THIS USEFUL? 👍 👎

◆ AI Chat Apps

Open WebUI

Feedback ⓘ v0.13.3 NOTES ↗ LATEST STABLE

Open WebUI is a self-hosted chat interface for local and remote models, with RAG over uploaded documents, tool calling, and multi-user access control.

Open WebUI adds coding agent support in automations and lets users switch terminals and working folders mid-chat

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/open-webui/open-webui:v0.13.3
```

pipx

```
$ pipx install open-webui==0.13.3
# already installed? upgrade to this version:
$ pipx install --force open-webui==0.13.3
```

git

```
$ git clone --branch v0.13.3 https://github.com/open-webui/open-webui.git
# already have the repo? check out this version:
$ git checkout v0.13.3
```

published under the project's GitHub account · PyPI attestation · checked 9 Nov Wrong command?

└──▷ HOW TO FIND IT

Run a coding agent on a schedule and have its output delivered directly to a chat instead of a channel.

📍 In the automation editor, go to the trigger settings and select a Terminal and working folder, then pick a coding agent listed under that terminal. Set your schedule, save the automation, and results will be posted to the designated chat.

Switch to a different project folder mid-conversation without starting a new chat.

📍 In an active chat (including a temporary chat), click the terminal or folder control in the chat header, choose a different terminal or folder, then continue sending messages — both controls are disabled only while a reply is generating.

Agents in automations: the automation editor now lets you choose a terminal and working folder, select a coding agent, and send scheduled run results to a chat.

Change the terminal or working folder after sending messages in a chat, including in temporary chats.

▾ **1 more release in this issue** · 2026-11-08

v0.13.1 NOTES ↗ STABLE

Open WebUI adds coding agents in the model picker, a desktop companion app (Tether), split-pane workspace, Git integration, and a raft of tool-permission and cost controls

└──▷ **GET THIS VERSION** ▾

Docker

```
$ docker pull ghcr.io/open-webui/open-webui:v0.13.1
```

pipx

```
$ pipx install open-webui==0.13.1
# already installed? upgrade to this version:
$ pipx install --force open-webui==0.13.1
```

git

```
$ git clone --branch v0.13.1 https://github.com/open-webui/open-webui.git
# already have the repo? check out this version:
$ git checkout v0.13.1
```

published under the project's GitHub account · PyPI attestation · checked 9 Nov

Wrong command?

└──▷ **HOW TO FIND IT**

Cap how far back a model can read through chat history to prevent runaway context costs in long-running conversations.

📍 In Admin Settings › General, set Chat Read Limit to your desired character cap — or set the environment variable `VIEW_CHAT_MAX_CHARS=20000` at startup. The newest messages are kept and older ones are trimmed to the limit.

Turn on human-readable labels for every tool call so non-technical users can see what a tool is doing without reading raw JSON.

📍 In Admin Settings › Interface, enable Tool Call Labels — or set `ENABLE_TOOL_CALL_LABELS=true` in your environment. The model will then generate a short description in the user's language for each tool call and its result.

Include small reference files in full (rather than chunking them) to improve answer quality when attaching config files, short docs, or code snippets.

📍 In Admin Settings › Documents, set the full-text threshold — or set `RAG_FULL_TEXT_CHAR_THRESHOLD=100000` to raise the ceiling to 100k characters. Files whose extracted text falls below the threshold are sent in full to the model.

Limit how much chat history the Chat History tools return by setting `VIEW_CHAT_MAX_CHARS` or the new Chat Read Limit field in Admin Settings > General, keeping the newest messages while older ones are cut.

Control full-text inclusion for small files with `RAG_FULL_TEXT_CHAR_THRESHOLD` (default 50,000 characters) or the admin Documents settings, so files below the threshold are included in full instead of through retrieval.

Enable readable per-call tool descriptions with `ENABLE_TOOL_CALL_LABELS` or Tool Call Labels in Admin Settings > Interface (off by default), asking the model for short labels in the user's language.

Pass the client IP address to upstream providers by using `{{USER_IP}}` or `{{X_FORWARDED_FOR}}` placeholders in custom connection headers.

Open WebUI adds Tether, a desktop companion app for macOS, Windows, and Linux that ties a local machine to Open WebUI over an outbound connection — no open ports required — with terminal output that resumes after dropped connections, desktop notifications, and a session expiry set via `SESSION_EXPIRES_IN` (default 90 days without use).

+5 more

└──▷ **BREAKING ON UPGRADE**

! Custom pipes see a different message layout: citation instructions, memory recall, the skills list, and attached knowledge are now delivered as hidden tool results rather than rewritten into the system prompt or user message on every tool round.

Our best guess

Check this claim Claim history

WAS THIS USEFUL? 👍 👎

◆ ML Platforms & SDKs

City2Graph

Feedback ⓘ v1.1.0 NOTES ↗

City2Graph builds graph representations of geospatial relationships for graph neural network and spatial network analysis.

City2Graph adds synthetic GTFS generation and hole filling for buffer-method isochrones

└──▷ GET THIS VERSION ▾

pip

```
$ pip install city2graph==1.1.0
# already installed? upgrade to this version:
$ pip install --upgrade city2graph==1.1.0
```

git

```
$ git clone --branch v1.1.0 https://github.com/c2g-dev/city2graph.git
# already have the repo? check out this version:
$ git checkout v1.1.0
```

named in the project's README · checked 9 Nov Wrong command?

Fill holes in isochrone polygons by passing the new hole-filling option to create_isochrone(method='buffer').

City2Graph adds synthetic GTFS generation and reusable benchmarks to support testing and performance measurement of transit graph workflows.

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

knowhere

Feedback ⓘ v1.2.47 NOTES ↗ LATEST STABLE

Knowhere extracts, parses, and outputs structured chunks ready for AI Agents and RAG.

knowhere now accepts TSV files for ingest alongside CSV and XLSX, with Gemini Embedding 2 token prices recorded for usage tracking

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.2.47 https://github.com/Ontos-AI/knowhere.git
# already have the repo? check out this version:
$ git checkout v1.2.47
```

Add `.tsv` to the `SUPPORTED_EXTENSIONS` environment variable on production and staging deployments to enable TSV file ingest.

knowhere now accepts TSV files for ingest, billed by visible non-empty cell count the same way CSV spreadsheets are.

knowhere now records Gemini Embedding 2 text and image token prices for usage tracking.

▾ 1 more release in this issue · 2026-11-09

v1.2.46 NOTES ↗ STABLE

knowhere adds direct tool invocation for host agents and Hipo via `POST /api/v2/retrieval/tools/invoke`, accepting guest API keys

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.2.46 https://github.com/Ontos-
AI/knowhere.git
# already have the repo? check out this version:
$ git checkout v1.2.46
```

└──▷ TRY IT

Call a corpus fuzzy-recall tool directly from an external agent using a guest API key, bypassing Knowhere's built-in retrieval agent.

```
$ curl -X POST https://<host>/api/v2/retrieval/tools/invoke \
  -H 'Authorization: Bearer <guest-api-key>' \
  -H 'Content-Type: application/json' \
  -d '{"tool": "corpus.recall", "corpus_id": "<corpus-id>", "query":
"safety shutdown procedures"}'
```

Call corpus tools `corpus.recall`, `corpus.grep`, `corpus.outline`, `corpus.read`, `corpus.query_table`, and `corpus.assets` directly via `POST /api/v2/retrieval/tools/invoke` without running Knowhere's own retrieval agent; guest API keys are accepted on this route, matching the access policy of `/v2/retrieval/query`, and tools outside the whitelist return `400`.

WAS THIS USEFUL? 👍 👎

Docling Graph

Feedback ⓘ v1.9.2 CODE ↗

Docling Graph converts documents into Pydantic objects and builds directed knowledge graphs with explicit semantic relationships using LLM or VLM backends.

└──▷ GET THIS VERSION ▾

pip

```
$ pip install docling-graph==1.9.2
# already installed? upgrade to this version:
$ pip install --upgrade docling-graph==1.9.2
```

git

```
$ git clone --branch v1.9.2 https://github.com/docling-project/docling-graph.git
# already have the repo? check out this version:
$ git checkout v1.9.2
```

PyPI attestation · checked 9 Nov Wrong command?

└──▷ BREAKING ON UPGRADE

! Node IDs change for entity identity values containing non-ASCII letters or script-specific marks (e.g. 'Łódź', 'Ølsen', CJK, Cyrillic, Greek, kana, Indic, Thai, Arabic, Hebrew names). Graphs built before v1.9.2 will not share node IDs with graphs built after it for those entities.

Our best guess

Check this claim Claim history

WAS THIS USEFUL? 👍 👎

garden-skills

Feedback ⓘ motion-showreel-v0.2.0 NOTES ↗

Garden-skills is ConardLi's open-source Skills collection, featuring web design, knowledge retrieval, image generation, and more.

garden-skills motion-showreel adds workspace preview step, confirmation sync workflow, shot browser toolbar, and three new visual styles

└──▷ GET THIS VERSION ▾

```
$ git clone --branch motion-showreel-v0.2.0
https://github.com/ConardLi/garden-skills.git
# already have the repo? check out this version:
$ git checkout motion-showreel-v0.2.0
```

Adds a workspace preview step before MP4 export, using `editor.mjs`, expanding the workflow from 7 to 8 steps with workspace preview as step 6 and export/acceptance as step 7.

Adds an optional `progress.mjs` for stage-by-stage progress updates when requested.

Adds a confirmation sync workflow with a `--confirm` flag to resolve waiting state before production work, including stale-status detection via file change monitoring with a 3-minute threshold, plus a `--reopen` flag required when revised results need re-review, preventing confirmed stages from accepting new waiting status without explicit reopen.

Adds a `stdoutFile` option to `runProcess` for full report capture while keeping the console buffer bounded.

Adds an optional render context that passes `frameTime` and `fps` to render when `RENDER_CONTEXT` is enabled, preventing subframe text blending at cuts.

+4 more

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

versus-incident

Feedback ⓘ v1.4.35 NOTES ↗

Versus-incident is the self-hosted AI SRE agent. You don't have to monitor your systems alone.

versus-incident adds configurable Kubernetes history retention, pageable cluster lists, and SQLite-backed Timeline and Changes views without Postgres

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/versuscontrol/versus-incident:v1.4.35
```

git

```
$ git clone --branch v1.4.35 https://github.com/VersusControl/versus-incident.git
# already have the repo? check out this version:
$ git checkout v1.4.35
```

published under the project's GitHub account · checked 9 Nov Wrong command?

└──▷ USE IT

Extend Kubernetes history beyond the new 7-day default to retain 30 days of cluster evidence.

config.yaml

```
kubernetes:
  evidence:
    retention: "720h"
    save_chunk: 500
```

Control how long Kubernetes cluster history is kept by setting `kubernetes.evidence.retention` in `config.yaml` (for example `720h` for 30 days, or `"0"` to keep everything); default is 7 days.

Control the save part size for large rollouts by setting `kubernetes.evidence.save_chunk` in `config.yaml` or via Helm.

Page through Kubernetes lists with range and page count displayed (for example `51–100 of 312 · Page 2 / 7`), newest breaks shown first, without missing or repeating rows.

See inline status messages (`History is temporarily not saving` or 'History could not save the latest data') when saving fails, with saved rows still visible.

See the cluster Overview on clusters where Versus cannot read some resource kinds (such as Nodes), with missing kinds listed.

└──▷ **BREAKING ON UPGRADE**

! History retention now defaults to 7 days. Set `kubernetes.evidence.retention` to `"0"` (keep everything) or a longer duration before upgrading to avoid data loss. During a rolling update, keep `"0"` until every replica runs v1.4.35.

Our best guess

Check this claim Claim history

! Page links saved before upgrading are invalidated; they restart from page 1. API clients receive `400 cursor_expired`.

Our best guess

Check this claim Claim history

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ updated · Nov 9, 2026 NOTES ↗

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

Braintrust adds thread translation (browser-native, no model credits), Terraform provider GA, trace continuation via `--parent` in `bt trace run`, ephemeral `braintrust_service_token`, `braintrust_users` data source, MCP `list_scheduled_loop_job_runs`, thread view in human review, GitHub integration for org-level Loop threads, and member 2FA visibility for org owners.

Continue a trace from a coding-agent run by passing `--parent` to `bt trace run` with the value returned by an SDK span's export(), attaching the session under an existing span instead of starting a new one.

Call a Braintrust MCP server tool with only its own arguments now that `bt mcp run` no longer requires the telemetry argument.

Manage saved views for Patterns as code now that `braintrust_view` accepts patterns as its `view_type` in the Terraform provider.

Create ephemeral service tokens without storing the key in Terraform state or plan files using the new `braintrust_service_token` resource.

Look up multiple organization members in one request using the new `braintrust_users` data source.

+8 more

NEW INTEGRATION Firebase Genkit tracing (Go SDK)

NEW INTEGRATION MCP server tracing for fastmcp and mcp-handler (TypeScript SDK)

UPDATED INTEGRATION Expanded framework tracing support (TypeScript SDK v3.43.0)

UPDATED INTEGRATION Anthropic Java SDK 3.x tracing support (Java SDK)

NEW INTEGRATION Temporal activity tracing (Python SDK)

└─▷ BREAKING ON UPGRADE

! The Terraform provider now requires Terraform or OpenTofu 1.11.1 or later.

Our best guess

Check this claim Claim history

! `braintrust_ai_secret` now requires `secret_version` alongside `secret`, and `braintrust_env_var` now requires `value_version` alongside `value`. Changing only the value no longer updates it; increment the version field to write a new value. References to `secret` or `value` return `null`.

Our best guess

Check this claim Claim history

! The `braintrust_org_automation` resource is removed because the Braintrust API no longer supports it. Delete its blocks from your configuration, then run `terraform state rm` on each address before running `terraform plan`.

Our best guess

Check this claim Claim history

WAS THIS USEFUL?  

GOVERN

◆ Model & Agent Protection

Minimal Software Research minimal

Feedback ⓘ

release-0bb305b9 NOTES ↗

Minimal is a secure, declarative runtimes for Claude Code, Gemini, Codex, Pi and more, so you can run multiple agents safely on your stack from a verified software supply chain.

minimal: Limit the number of registered client keys on the Box Egress Proxy with `--max-client-keys` on `bep serve`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch release-0bb305b9
https://github.com/gominimal/minimal.git
# already have the repo? check out this version:
$ git checkout release-0bb305b9
```

└──▷ TRY IT

Store a GitHub token as a named secret inside a box so the agent can use it without you pasting it into the session.

```
$ min secret set GH_TOKEN
```

Inspect the full audit trail for a box, merging its parent chain, to verify no egress decisions were tampered with.

```
$ min box audit --parent mybox
```

File a suspected security vulnerability in the tool directly from the CLI, bundling the telemetry spool for context.

```
$ min bug --security
```

Sign in to GitHub through `min login`, completing sign-in on a real page with repository choice; logging out deletes the sign-in and refuses every reference to it.

Keep store rules under `min --config-dir` in step with `bep serve` for consistent egress policy.

Check once a day whether the channel has a newer release and report it via `min`.

Export VM host daemon spans and carry telemetry into the guest via `minvmd`.

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Open-Weight Models

OmniJev OneJev1.6g-31B

Feedback ⓘ released · Nov 9, 2026

MODEL ↗

OneJev1.6g-31B from OmniJev was released on 8 Nov 2026. Open weights on Hugging Face.

WAS THIS USEFUL? 👍 👎

H2O.ai h2o-lightning-12b

Feedback ⓘ

released · Nov 9, 2026 MODEL ↗

h2o-lightning-12b from H2O.ai was released on 8 Nov 2026. Open weights on Hugging Face.

WAS THIS USEFUL? 👍 👎

Z.ai GLM 5.3 Flash

Feedback ⓘ price cut · Nov 8, 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 → $0.0335 in, $0.50 → $0.083 out per million tokens (-83%), first seen 8 Nov 2026.

WAS THIS USEFUL? 👍 👎

Moonshot AI Kimi K2.5

Feedback ⓘ

retirement scheduled · Nov 8, 2026 MODEL ↗

Kimi K2.5 is a multimodal model from Moonshot AI.

OpenRouter lists a retirement date of 18 Nov 2026 for moonshotai/kimi-k2.5, first seen 8 Nov 2026.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

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

▷ Subscribe Feedback on this issue