THE AI TOOLCHAIN

NO. 073

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED OCTOBER 25, 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 1993 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 247 of 253 tools

OFFENSIVE SECURITY

vim ? help

◆ Exploitation & C2

xalgorix

Feedback ⓘ v4.6.164 NOTES ↗ RANK

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

> xalgorix shipped a cluster of opt-in environment-variable controls for scan termination, token efficiency, and model tool-calling compatibility, plus a target-wide finding deduplication pass that collapses repetitive issue classes into single findings.

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

01 **Native structured tool calling for local model backends** NEW

83

Adds `XALGORIX_NATIVE_TOOL_CALLS` environment variable (default off) to enable a native structured tool-calling channel that sends tool definitions as a structured array and reads model tool calls back as structured data, bypassing the XML tag path. Supported on llama.cpp, Ollama, and vLLM backends.

— Named flag, default, and three supported backends with mechanism v4.6.163

02 **Stuck-task advance recovery** NEW

75

Adds `XALGORIX_ADVANCE_PAST_STUCK` env var (default off) to skip un-completable tasks and continue scanning, bounded by `XALGORIX_MAX_STUCK_ADVANCES` (default 3) consecutive no-progress advances.

— Two named env vars with defaults and bounding mechanism v4.6.160

03 **Plan-complete idle cap** NEW

75

Adds `XALGORIX_PLAN_COMPLETE_IDLE` environment variable (default `600`) to finalize scans after a bounded no-new-findings tail once the structured plan is exhausted — any new finding resets the counter.

— Named env var, default value, and reset mechanism given v4.6.159

04 **Bounded epoch snapshots for large-surface scans** NEW

72

Adds `XALGORIX_EPOCH_COVERAGE` (default off), an opt-in flag that freezes the attack-surface snapshot into bounded epochs so per-endpoint coverage obligations can actually complete instead of growing indefinitely as reconnaissance discovers new endpoints.

— Names the flag and default, mechanism of bounded epochs explained v4.6.164

05 **Target-wide finding deduplication** IMPROVED 70

Adds a `debug_disclosure` deduplication class (stack-trace / debug-page / verbose-exception keywords; CWE-209) and a `user_enumeration` deduplication class (CWE-203/204) that collapse error-verbosity and account-enumeration findings instead of re-reporting per path. Also collapses server-wide issue classes — debug/stack-trace disclosure, missing security headers, CORS policy, version/technology disclosure, and clickjacking — to one finding per target instead of re-reporting on every path.

— Names every dedup class and CWE but behavior is automatic, not user-triggered v4.6.160

06 **Lean tool schema for smaller prompts** NEW 65

Adds `XALGORIX_LEAN_TOOL_SCHEMA` env var (default off) to strip illustrative example spans from tool/parameter descriptions, shrinking per-request prompt size while preserving all tools, parameters, and core instructions.

— Named env var and default, mechanism of stripping spans given v4.6.160

THINNER COVERAGE BELOW

07 **Malformed tool-call tag recovery** IMPROVED 47

Recovers tool calls where a model emits `<parameter=NAME>` instead of `<function=NAME>`, avoiding the wasted retry turn and full context resend.

— Names the two tag formats but is an automatic fix, not user-facing v4.6.159

WAS THIS USEFUL? 👍 👎

APPLICATION SECURITY

◆ SAST / SCA / Code Security

jscpd

Feedback ⓘ v5.4.2 NOTES ↗

Copy/paste detector for source code. 220+ languages, Rust engine, SARIF/HTML/badge reporters, GitHub Action, MCP server for AI agents.

jscpd 5.4.2 adds module boundaries enforcement, consistent identifier mapping for renamed-clone detection, and git history health charting with `--history-every`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v5.4.2 https://github.com/kucherenko/jscpd.git
# already have the repo? check out this version:
$ git checkout v5.4.2
```

Adds `boundaries` key to `.jscpd.json` with `code`, `notIn`, `allow`, and `thresholds` fields; violations exit with code 1 and appear under `boundaries` in the JSON report.

Adds `--ignore-identifiers=consistent` for one-to-one name mapping in renamed-clone detection; `--ignore-identifiers=blind` restores the previous any-to-any behavior.

Adds `--history-every N` and makes bare `--history` walk the last 30 commits (`--history-limit`); console gains HEALTH and CHANGE columns and a health chart; JSON gains a `health` field per commit point.

Improves `--similarity` fingerprinting so each renamed local maps to exactly one counterpart, eliminating false high-similarity scores between structurally different functions.

NEW SUPPORTED PLATFORM 73 new file format mappings including Dockerfile, Makefile, go.mod, Elixir, Nix and more

└──▷ BREAKING ON UPGRADE

! A baseline made with `--ignore-identifiers` in v5.4 is treated as blind; use `--ignore-identifiers=blind` or regenerate the baseline to avoid false `--fail-on-new-clones` failures.

! `jscpd .` now scans additional files due to new format mappings, which can shift duplication percentage and trigger a `--threshold` failure.

! `--history` now walks only first parents, producing fewer data points across merged branches; the ai reporter line format changed.

WAS THIS USEFUL? 👍 👎

BUILD

◆ AI Coding Agents

StackBlitz bolt.new

Feedback ⓘ snapshot-20261025 NOTES ↗

RANK

Bolt.new is an AI-powered web development platform that builds websites and applications from prompts.

> Bolt.new added GitHub-linked skill syncing, letting imported skills be refreshed from their source repo, while introducing a breaking change where editing such a skill severs that link.

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

01 **Manual update for GitHub-imported skills** NEW 65

GitHub-imported skills now stay linked to their source repository folder. Users can pull the latest version via `Update from GitHub` in the skill's menu, which replaces the existing skill with the latest version from its source repository.

Pull in the latest version of a GitHub-linked skill after its source repo has been updated.

📍 Open the skill's menu and click Update from GitHub to replace the existing skill with the latest version from its source repository.

– Names exact menu action and behaviour, no API/flag given snapshot-20261025

THINNER COVERAGE BELOW

02 **Editing disconnects skill from GitHub** BREAKING 45

Editing a GitHub-imported skill disconnects it from GitHub, turning it into a standalone local copy that can no longer be synced or updated from the source repository.

– Clear before/after behaviour but no workaround or command given snapshot-20261025

└──▷ BREAKING ON UPGRADE

! Editing a GitHub-imported skill disconnects it from GitHub, making it a standalone local copy with no further sync capability.

WAS THIS USEFUL? 👍 👎

claude-mem

Feedback ⓘ v15.2.0 NOTES ↗ RANK

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

> claude-mem shipped a new browser-based install flow that extracts Work, Goals and Rules from local agent history, added multi-agent support (Claude Code, Codex, Gemini) for its refresh step, replaced its static viewer with a live React My Work dashboard, and added opt-in MCP loopback transport plus a Docker image improvement.

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

01 Browser-based install flow with hosted extraction NEW

85

`npx claude-mem install` now triggers a browser setup flow that scans the last 30 days of local agent history, deduplicates and redacts prompts, and uploads filtered prompts (gzip) to CMEM Cloud for intent extraction, populating local My Work snapshots with returned goals, rules and request groups before setup completes. Conversations the hosted scan could not group are now saved as undecided My Work requests instead of aborting setup with a `Hosted extraction omitted conversation ...` error, so the local refresh can group them later.

— Names exact command and upload mechanism; no control over the new fallback behavior given v15.2.0

02 Live React My Work viewer with real-time agent status NEW

78

Replaces the static localhost viewer at `http://localhost:37777` with a live React My Work viewer showing one card per human request, progress meters, owner goals, goal alignment, shipping velocity, and the build story; adds a Classic viewer at `/viewer` as a linked fallback from the new My Work header; changes now appear in real time without a manual `Show updates` step; and adds per-agent refresh failure visibility, finished-subagent status display, and the ability to mark a subagent finished from its agent card.

— Exact URLs and UI elements given, but behavior otherwise descriptive v15.1.0

03 Multi-agent refresh profiles for Claude Code, Codex and Gemini NEW

65

The goals-and-progress refresh step now supports Claude Code (bounded file tools and memory recall only), Codex (strict read-only workspace/runtime profile, no writes, no network, native and npm runtime layouts), and Gemini (prepared evidence inline, no builtin tools, hooks or YOLO mode) as refresh agents.

— Names three agent profiles and their constraints but no invocation command v15.2.0

THINNER COVERAGE BELOW

04 **Opt-in loopback HTTP transport for MCP** NEW 35

Adds opt-in loopback HTTP transport for MCP with per-session workspaces.

— No flag or config key given to enable this transport v15.1.0

05 **Embedding model baked into Docker image** IMPROVED 25

Bakes the configured embedding model into the Docker image.

— Single sentence with no configuration or version detail v15.1.0

WAS THIS USEFUL? 👍 👎

claude-council

Feedback ⓘ v2026.10.12 NOTES ↗ RANK

Claude Code plugin that queries multiple AI coding agents and shows their answers side by side. Supports Gemini, OpenAI, Grok, Perplexity, Kimi, and OpenRouter models.

> claude-council added an opt-in heads-up side-check that surfaces mid-turn notes from a council seat, redesigned the in-editor council pane with per-seat rows and a retry button, and added OpenCode CLI and GitHub Copilot CLI as new council seats — while migrating the Perplexity seat to its Agent API and removing the specialists feature entirely.

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

01 **Heads-up side-check for council seats** NEW 93

Adds `/heads-up setup`, `/heads-up on [seat]`, `/heads-up off`, and `/heads-up` subcommands, plus a `.claude/council-heads-up.json` project config supporting `enabled`, `seat`, and `every` (checks per model request, default 3); can also be enabled globally via the Heads-up side-check row in `/config`. It sends the last five turns and last eight tool results to one council seat mid-turn, surfaces a tagged Heads up or You should know note above the prompt, and offers interactive controls `1: Learn more`, `2: Knew this already`, `4: Turn off` (double-press writes project file off), `0: Dismiss`, and `1: Tell Claude`, reporting bytes sent and the answering seat. A later release adds a separate `1: Understood` option counted apart from Knew this already, and restructures the Tell Claude draft into a headline, a bulleted list, and a closing prompt asking Claude to verify the claim before acting. Another release passes digest slot data to `jq` via a file instead of argv, lifting a 32 KB argument limit on Windows so the check now receives the full text of Write/Edit tool results up to 2,000 characters per slot, and adds `--tool-results K` and `--tool-result-chars N` flags to `transcript-digest.sh`.

Turn on the heads-up side-check for a single project and set it to check every 5 model requests using the codex seat.

```json
{ "enabled": true, "seat": "codex", "every": 5 }
```

Enable the heads-up check interactively and choose which seat, model, and cadence it uses.

```
$ /heads-up setup
```

Include the last 10 tool results (capped at 2000 chars each) in the transcript digest to give council seats more tool context.

```
$ transcript-digest.sh --tool-results 10 --tool-result-chars
2000
```

— Full mechanism plus exact commands, config keys and flags given. v2026.10.10 v2026.10.9 v2026.10.6

02 **OpenCode CLI as council seat** NEW 86

Adds `opencode` as an opt-in council seat, enabled via `--providers` or `COUNCIL_PROVIDERS`, with its model sourced from `opencode.json` and overridable via `OPENCODE_MODEL`. `COUNCIL_PASS_ENV` selectively forwards environment variables to the seat when vendor API keys and other `OPENCODE_*` variables are otherwise blocked; each run gets a private server and empty config dir excluding the user's config, plugins, MCP servers, project `AGENTS.md`, and tools, and deletes its own session from OpenCode history afterward. `/claude-council:status` gets an OpenCode row showing `Connected` or prompting `opencode auth login`.

Add OpenCode to a one-off council question alongside another provider, overriding the model it uses.

```
$ OPENCODE_MODEL=claude-sonnet-4-5 /claude-council:ask --
providers opencode,codex "Which auth strategy is safer for a
public API: JWT or opaque tokens?"
```

Pin OpenCode in every council run for a project and pass through a specific env var it needs.

```
$ COUNCIL_PROVIDERS=opencode,codex COUNCIL_PASS_ENV=MY_ORG_TOKEN
/claude-council:ask "Review this service boundary for coupling
issues."
```

— Named flags, env vars and runnable example commands provided. v2026.10.12

03 **GitHub Copilot CLI as council seat** NEW 83

Adds a `copilot-cli` seat that places GitHub Copilot CLI on the council, activated by naming `copilot-cli` in `--providers` or `COUNCIL_PROVIDERS` (never auto-seated, since each question spends a premium request). It invokes Copilot as `copilot --output-format json --no-auto-update --no-custom-instructions --disable-builtin-mcps --available-tools=none` with the prompt on stdin, giving the seat no tools and no `AGENTS.md`, and scopes the env scrub to `COPILOT_GITHUB_TOKEN`,

`COPILOT_HOME`, `COPILOT_CLI_*`, and gh config/host variables, explicitly excluding `GH_TOKEN`, `GITHUB_TOKEN`, and `COPILOT_ALLOW_ALL`. `/claude-council:status` now reports whether `copilot` is on `PATH`.

Pin `copilot-cli` permanently in your shell environment so every council run includes it without repeating the flag.

```
$ export COUNCIL_PROVIDERS=copilot-cli,codex,gemini
```

— Exact invocation flags and env scope given, plus a pin command. v2026.10.5

---

04 **Perplexity seat migrated to Agent API** BREAKING 78

The Perplexity seat now posts to `/v1/agent` with the `web_search` tool instead of `/chat/completions`. `PERPLEXITY_MODEL` accepts presets (`fast`, `low`, `medium`, `high`, `xhigh`) or a `provider/model` id, defaulting to `low` with fallback `fast`; legacy `sonar-*` ids are refused before the first request — replace with `fast` for `sonar` / `sonar-pro`, `low` for `sonar-reasoning-pro`, or `high` for `sonar-deep-research`.

— Exact endpoint and migration mapping given, no runnable command. v2026.10.6

---

05 **Redesigned council pane with per-seat controls** IMPROVED 70

Redesigns the in-editor council pane so each seat occupies one row showing provider edge, name, model, a state bar, elapsed time, and a context-sensitive action button (`[ cancel ]`, `[ retry ]`, `[ open ]`, `[ fold ]`), with a footer offering fold/expand and cancel all controls. The status band now uses plain-text indicators: `● running COUNCIL` with sweeping dash, clock, `N of M done · <latest event>`, and terminal states `✓ done`, `✗ failed`, `✗ retry`; the heads-up note badge is standardised to a `!` or `?` mark in white on orange, one cell wide on every terminal. A later release adds a `retry` button for a seat that fails while others are still running — it replaces `cancel`, re-queries the seat once, shows `retrying` until restart, and only appears while the watch directory's `retriable/` folder marks that seat as retryable.

— Named UI states and buttons but no command to invoke, just UI behaviour. v2026.10.11 v2026.10.7

THINNER COVERAGE BELOW

---

06 **Specialists feature removed** BREAKING 50

Removes the `/specialists` screen, the `specialist` tool, `scripts/specialist.sh`, the six bundled template skills, the `specialists` and `specialist_round_limit`

config settings, and the `.worktreeinclude` file; any configuration or scripts referencing these will break on upgrade.

— Names every removed surface but gives no migration path. v2026.10.7

---

07 **Synthesis header in saved transcripts** NEW 35

Adds a `## 🧭 Synthesis` header, preceded by a rule, to the saved transcript and Claude's reply, visually separating the synthesis from individual seat answers.

— Names the exact header markup but is a one-line cosmetic change. v2026.10.11

┗--▷ BREAKING ON UPGRADE

! The `/specialists` screen, `specialist` tool, `scripts/specialist.sh`, the six bundled template skills, and the `specialists` and `specialist_round_limit` config settings and `.worktreeinclude` are removed; any configuration or scripts referencing these will break on upgrade.

! The Perplexity seat now posts to `/v1/agent` instead of `/chat/completions`; any `PERPLEXITY_MODEL` set to a legacy `sonar-*` id is refused before the first request — replace with a preset: `fast` for `sonar` / `sonar-pro`, `low` for `sonar-reasoning-pro`, `high` for `sonar-deep-research`.

WAS THIS USEFUL? 👍 👎

coffee-cli

Feedback ⓘ v3.6.9 NOTES ↗ RANK

A clean AI terminal workspace. No hacks, no extra hooks, no lock-in — with quick input, folder switching, copy & paste, and image pasting. 极致清爽的 AI 终端工作平台。0 魔改、无额外 Hook、无绑定；支持快捷输入、目录切换、复制粘贴与图片粘贴。

> Coffee CLI's biggest changes this window are Coffee Agent gaining full remote control of desktop terminal sessions via WeChat and Telegram, four new proactive work modes (Sentinel, Project Manager, Expert Configuration, English Delegation), and new desktop sessions, split panes, and usage statistics for Claude Code and Codex; launch support also expanded to Kiro, MiniMax Code, and Muse Code.

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

01 Launch command support for new tools and presets NEW

95

The `coffee-cli launch --tool <id> [--cwd <dir>]` command opens a new agent tab directly from launchers, context menus, and shell aliases, forwarding to a running instance if one exists — e.g. `coffee-cli.exe launch --tool claude --cwd "C:\work\project"` or `coffee-cli launch --tool muse --cwd ~/work/my-project`. Launch support now covers Kiro and `MiniMax Code` with brand icons, and `muse` (Muse Code, using the Meta PNG icon, with Windows launcher system paths no longer overwriting directory-based tab names), plus editable launch command presets for Claude Code and Codex accessible from tool card gear icons.

Open a Claude Code tab on Windows pointed at a project folder, e.g. from an Explorer right-click entry.

```
$ coffee-cli.exe launch --tool claude --cwd "C:\work\project"
```

Launch Muse Code directly into a project directory from a shell alias or context menu entry.

```
$ coffee-cli launch --tool muse --cwd ~/work/my-project
```

— Runnable command plus every named launch target v3.6.0 v3.5.9 v3.6.1 v3.6.9

02 Coffee Agent remote control via WeChat/Telegram NEW

81

Coffee Agent lets users delegate and control desktop CLI terminal sessions from WeChat or Telegram: connect via a WeChat QR code or Telegram Bot pairing, pick a tool and working directory, dispatch tasks through Gambit, and resume saved sessions, using a

separately configurable model. Conversations persist locally for cross-device continuity — summarizing long conversations and remembering, reviewing, and forgetting work preferences — with WeChat/Telegram threads reopenable from desktop session history, task delegation and progress reporting for tools like Claude Code and Codex, a Coffee Agent UI available in 11 languages, and a bubble conversation view with per-message copy buttons. Desktop support now also lets users read, send, and interrupt within the original session without opening a separate terminal.

— Rich mechanism and named surfaces, but no exact command shown v3.5.7 v3.5.8 v3.5.9 v3.6.9

03 **Character Flow background animation** NEW 80

A new Character Flow appearance option (Settings → Appearance → Shape) renders drifting background glyphs tinted to the active theme colors, reacting to tab switches across Terminal, Coffee Agent, message settings, and Diff tabs with a ripple from the selected tab; all appearance Shape names, including Monogram and Character Flow, are localized across all 11 interface languages.

Enable the new Character Flow background so glyphs drift behind your workspace in your current theme's colors.

📍 In the app, go to Settings → Appearance → Shape and select Character Flow.

— Exact settings path plus a usage example v3.6.4

04 **OpenCode v2 support** NEW 70

Coffee CLI now supports OpenCode V2 session history and heatmaps, and adds Dynamic Island support for OpenCode v2, displaying working, waiting-for-permission-or-answer, and idle states including subtask activity, with stale status hidden on disconnect and no changes required to OpenCode's global configuration.

— Single-line mention with no mechanism or scope Detailed states and behavior but passive feature, no action needed

v3.5.7 v3.6.2

05 **General settings with interface scaling** IMPROVED 70

A new General settings section consolidates interface scale (70%–150%) and task style as the first settings item, with restore-default and restart-persistent preference; tick marks and solid thumbs are unified across the interface-scale and wallpaper-opacity sliders, with scale previewing while dragging and applying on release, and opacity updating live with fewer redundant refreshes.

— Specific ranges and behavior, settings location given v3.6.6

06 **New work modes for Coffee Agent** NEW 70

Adds four new work modes: Sentinel mode, which proactively pushes task results through connected WeChat or Telegram channels after observing task completion, with explicit reporting of interruptions and incomplete output (initial support covers Claude Code, Codex, Kimi Code, and Oh My Pi); Project Manager mode, which reads project files, recent Git commits, and uncommitted changes, stores project understanding and memory, clarifies ambiguous requirements before delegating, and optionally integrates with Graphify to extract code structure and relationships, saving scan data to Coffee CLI's local data directory; Expert Configuration mode, which inspects existing Skills and MCP settings, suggests project-relevant candidates, and after confirmation installs project-level Skills or supported MCPs for Claude Code and Codex in visible terminals, verifying results and attempting rollback on failure; and English Delegation mode, which accepts tasks in the user's language, delegates to other CLI agents in English, and reports results back in the user's language — all grouped in Message Settings via a two-by-two switch layout with explanatory tips and automatic saving.

— Names supported tools and channels but no setup steps Names Graphify integration and storage but no how-to steps Clear workflow with verification and rollback described Thin description with no mechanism or scope detail Names the UI location but is purely a layout change

v3.6.0

07 **Local usage statistics view** NEW 60

A new usage statistics view shows tokens, sessions, trends, and model distribution, filterable by date, tool, and project, with cached results displayed first while a background sync runs.

— Names filters and behavior but not a navigation path v3.6.9

THINNER COVERAGE BELOW

08 **Dynamic split panes in session list** IMPROVED 55

Panes can now be opened or closed directly from the session list using numbered panels and hover-to-close controls, replacing the previous fixed two-, three-, and four-pane launchers.

— Clear before/after but no exact UI steps v3.6.9

09 **In-app release notes** NEW 55

Release notes now display automatically once after installation or upgrade, remain accessible from the top-right corner, follow the interface language, support local caching and offline reading, and sync automatically to the remote version manifest.

— Gives exact UI location and behavior v3.6.9

10 **Desktop sessions for Claude Code and Codex** NEW 50

Adds desktop-style sessions for Claude Code and Codex including chat, tool execution, permission requests, history navigation, and welcome animations; Desktop is now the

default experience for new users.

— Describes scope but no exact navigation given v3.6.9

11 **Durable agent memory with background extraction** NEW

50

Background extraction captures lasting preferences and user-confirmed project decisions, with identity-scoped recall, deduplication, updates, and capacity management.

— Explains mechanism but no interface or command shown v3.6.1

12 **Recoverable output compression** NEW

50

Long history and project results now use compact, RTK-style summaries with exact paths and identifiers, with saved originals readable page-by-page after restarts or new conversations.

— Describes mechanism but passive, no user action needed v3.6.1

13 **Chunked sending for long prompts** IMPROVED

45

Long prompts are now sent in ordered chunks (community contribution #153), preserving emoji boundaries and suppressing automatic submission after a write failure.

— Specific mechanism but no way to invoke it directly v3.6.6

WAS THIS USEFUL? 👍 👎

yetone magpie

Feedback ⓘ v0.1.1535 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's biggest addition this window is client-side privacy masking that redacts secrets and PII from prompts before they ever reach an AI vendor, backed by a new offline evaluation harness for measuring redaction accuracy; it also adds time-zone aware routing windows, a resident-memory budget for Claude Code sessions, and several smaller CLI and skill-discovery fixes.

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

01 Client-side privacy masking for prompts

UPDATED DETECTION NEW

93

Masking replaces API keys, private keys, tokens, emails, card numbers, and ID numbers with placeholders like `{{API_KEY_k3v9x2mq}}` on the local machine before any request is sent to an AI vendor, restoring real values in streamed replies and tool-call arguments. A local on-device NLP model (1.6 GB, CPU-only) catches names, addresses, and passwords that rule-based detection misses. Per-kind controls cover categories including passport, IBAN, MAC address, and public IP, each with a Mask, Warn, or Block action, plus a Never mask exemption list for values agents need to reason about directly. A `DO_NOT_TRACK=1` environment variable disables telemetry reporting, an Insights › Masking view shows every altered request side-by-side with what the vendor actually received, and a Try it panel in Settings previews masking on sample text before enabling it.

Prevent secrets in `.env` files or logs from reaching AI vendor servers by enabling privacy masking, then opting out of telemetry on a CI machine.

```
$ DO_NOT_TRACK=1 magpie
```

— Names env var, placeholder format, UI panels, and exact categories. product docs

02 Redaction evaluation harness for masking accuracy

NEW

85

Adds a redaction evaluation harness (`internal/redact/eval_test.go`) that scores three masking modes — rules-only, model-only, and combined — against a corpus of realistic agent texts (prose, code, logs, tool results, JSON), caching results in `testdata/eval-model.json` so the suite runs offline; set `MAGPIE_EVAL_CACHE` to re-run against the live local model. Eval cases were extended to correctly classify CI table

run IDs and filenames, git-log commit hashes and version strings, pixel-dimension strings, AI assistant model names (Claude, Codex, Opus, Sonnet, Gemini), and commit author logins and full SHA hashes.

Re-run the redaction evaluation against the live local model after adding new eval cases, pointing at the folder that holds the downloaded model cache.

📍 `MAGPIE_EVAL_CACHE`=$HOME/.cache go test . `/internal`/redact/...

— Names test file, cache file, env var, and a runnable command. v0.1.1533 v0.1.1518

---

03 **Resident-memory budget for Claude Code sessions** NEW

83

Adds a resident-memory budget (`claudeRunBudget` , floored at `claudeRunFloor` = 1 GiB, defaulting to `claudeRunUnknown` = 4 GiB when limits are unknown) so that a Claude Code run holding more than an eighth of the host's available memory after its turn is let go instead of kept, with its session shelved and resumed in a fresh process on the next turn.

— Names exact budget fields and thresholds, no direct command. v0.1.1519

---

04 **Clearer MCP library add-without-agents messaging** IMPROVED

80

Running `magpie library mcp add` without `agents=` now clearly reports the server is in the library but given to no agent yet, and prints the follow-up `magpie library mcp agents <name> <a,b…>` hint instead of the previous misleading 'every agent already has it' message.

Add an MCP server to the library without assigning it to any agent yet, then wire it up later.

```
$ magpie library mcp add ctx7 https://mcp.example.com/mcp
# Output: ✓ ctx7 is in the library, given to no agent yet
#   magpie library mcp agents ctx7 <a,b…> gives it to them
magpie library mcp agents ctx7 claude,codex
```

— Exact commands and output text given, directly runnable. v0.1.1535

---

05 **Time-zone aware routing time windows** NEW

75

Adds a `TimeWindow.Zone` field to routing rules so that time-window hours are evaluated in the zone where the rule was configured rather than on the server's local clock — a rule for vendor peak-price hours now stays anchored to those hours regardless of where magpie runs.

— Names the exact config field and the before/after behavior. v0.1.1524

THINNER COVERAGE BELOW

06 **QwenWork quota display via updated auth plugin** IMPROVED

55

Bumps the minimum version of the `@magpie-community/opencode-qoder-auth` plugin to `0.2.37` so that QwenWork personal-account usage displays remaining quota.

— Names plugin and version but gives no usage steps. v0.1.1518

07 **magpie-setup promoted in market skill search** IMPROVED

45

Promotes `magpie-setup` to first position in the market skills list so agents can find and install it when searched by terms like `setup`, `routing`, or `gateway key`.

— Names the skill and search terms but no mechanism. v0.1.1535

NEW INTEGRATION Antigravity CLI (agy) session integration

UPDATED DETECTION CJK and mixed-script secret-name vocabulary expansion

NEW INTEGRATION AionUi integration

v0.1.1526 CODE ↗ STABLE

Magpie v0.1.1526 surfaces Antigravity CLI (agy) sessions — with title, folder, token counts, and resume — on the Sessions page.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.1526 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.1526
```

Adds `agy` as a supported agent ID in the `agent` field of the `Session` JSON object, enabling Antigravity CLI conversations to appear on the Sessions page with title, folder, token counts, and resume.

Adds `agyCarry` state tracking in `sessions.go` to record where each agy transcript was read to, enabling incremental reads of `brain/<conversation-id>/.system_generated/logs/transcript_full.jsonl`.

v0.1.1510 CODE ↗ STABLE

Redaction engine extended to mask digit-only secrets and new CJK/Latin name patterns, while preserving numeric unit expressions like `2048 位`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.1510 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.1510
```

Extends the `SECRET` redaction rule to mask digit-only values (six or more digits) following CJK key-name triggers via `为`, `是`, `：`, `:`, or `=`, while leaving unit expressions like `2048 位` unmasked.

Adds `PIN码`, `安全码`, `授权码`, `暗証番号`, `비번`, and `pin` / `pin code` to the secret-name vocabulary so they are now redacted alongside existing key and token names.

v0.1.1509 CODE ↗ STABLE

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.1509 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.1509
```

WAS THIS USEFUL? 👍 👎

mvschwarz OpenRig

Feedback ⓘ v0.7.1 NOTES ↗ RANK

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

> OpenRig shipped eight releases in this window, headlined by per-seat typing guards and terminal policies, an HTTP-reachable MCP server and ACP agent dispatch, live edge editing on running rigs, a sweep of reliability work across Slack delivery, queue verification, rig send, and restore/launch state handling, plus new per-seat reasoning-effort controls and cost tracking.

└──▷ **WHAT SHIPPED · 42 FEATURES** most completely described first ? what's the number?

01 **Seat typing guards and terminal policies** NEW 91

`rig seat set-typing-guard <seat> --mode off|openrig-urgent|urgent-only|hold` controls how much automatic input reaches a seat, with `rig send --urgent` marking a message urgent; `rig seat set-terminal-policy <seat> --preset no-peer-interventions|help-within-instructions|full-control` sets how far OpenRig and other agents may act on a seat; `rig seat status` shows the permissions a seat's next fresh launch will use and where each comes from.

— Exact flags and modes named; no end-to-end usage walkthrough given. v0.7.1

02 **Slack delivery reliability, recovery and notification scoping**

IMPROVED 85

`rig gateway human set <id> slack-subscription urgent-only` limits Slack notifications to asks marked `--alert`; `queue.wake_human_rung` (default `always`, or `explicit-only`) keeps agent-to-agent escalations off a person's Slack unless the row is addressed to a person. Slack missed-message recovery was extended to mapped channels, replies in tracked threads whose root was posted or closed within seven days, and hub-relayed channels on machines without their own Socket Mode connection, and `rig slack status` gained per-channel/thread recovery lines plus a `Delivery:` line showing socket state and a retry-backlog count of retained inbound messages, reactions and click-answers. `rig slack enable` now reports how many earlier posts are waiting to retry, keeping and retrying failed first-post alerts on next gateway start while suppressing alerts for since-closed rows. Typed replies to Slack decisions with structured questions are now recorded on the row in `humanAnswers`. `rig queue create --verify` to a person who is also a seat now reports the Slack receipt in a `gateway` object (`posted`, `transport-failed`, `never-posted`, `still-pending`), and `GET /api/queue/:qitemId` returns `ownerNotificationLevel`. `rig slack verify` no longer accepts `--reason` or `--actor` flags.

— Many named fields and endpoints, but spans several loosely related fixes. v0.7.1

v0.6.13 v0.6.11 v0.6.10 v0.6.9

03 **Seat restart for Codex on Linux** NEW 84

`rig seat restart <seat> --keep-conversation --reason "<why>"` restarts a Codex seat in its own pane on the same conversation on Linux, with `--plan` to dry-run all checks and `--verify --expect-account <account>` to confirm the account after restart.

— Full command with flags and verification step given. v0.6.11

04 **Compact output modes for queue commands** NEW 82

`output.compact` (or `OPENRIG_OUTPUT_COMPACT`) makes the seven queue write commands (`rig queue create`, `claim`, `unclaim`, `update`, `block`, `handoff`, `handoff-and-complete`) print a short receipt instead of the full row, with `--full` printing the whole row; compact `rig queue list` rows (also `overdue` and `undelivered`) now omit absent fields entirely instead of returning `null` or `""`, adding a `fieldsElided` field naming what was dropped.

— Named config key, env var and all seven affected commands. v0.6.11 v0.6.10

05 **rig send reliability and delivery confirmation** IMPROVED 82

`rig send --wait-for-idle N` now waits before taking the seat's delivery lease (so other sends and queue wakes aren't blocked behind a long wait) and treats N as a

deadline from the start of the wait rather than from lease acquisition. `rig send <session> --host <id>` to an http-registered host now uses a 30-second request budget (plus any `--wait-for-idle` time) instead of 5 seconds, reporting `Delivery UNCONFIRMED` (`failedStep: 'remote-outcome-unknown'`, exit 1) when the outcome can't be confirmed, and directing the user to verify with `rig capture <session> --host <id>`. Codex seats behind a launcher now receive `rig send` and queue wakes again (broken since 0.5.17), and sends to Claude seats showing a faint suggested prompt in an empty input box now go through.

— Exact timings, exit codes and verification command given. `v0.6.11` `v0.6.10` `v0.6.9`

---

06 **Seat CWD move** NEW 81

`rig seat set-cwd <seat> --cwd <absolute-path> --reason <text>` moves an existing seat to a new working folder without losing its address, queue, or history, with conflict detection against startup files.

— Exact command, flags and conflict behaviour given. `v0.6.13`

---

07 **Quiet hours and person modes for gateway humans** NEW 80

`rig gateway human quiet set <id> --daily HH:MM-HH:MM` (or `--until`) holds routine asks during a person's away time, posting each once when they return while letting escalations through. `rig gateway human mode` saves and applies person modes with `--until` scheduling, delegated decisions inside a mode's agreed `decision room`, and a report of what each mode changed; `rig gateway human list` output also gained an `availability` field (`available|focus|away|off`).

— Named commands and fields, scheduling mechanism described. `v0.7.0`

---

08 **Intentional-stop tracking for rig start and rig ps** IMPROVED 80

`rig start --last` now leaves intentionally-stopped rigs down and prints `rig start --rigs <rig>` to bring them back, with `keptDown` in `--json` output; `rig seat stop` now records intentional stops (via migration `103`'s `intentionally_stopped` column) so a deliberately stopped seat no longer marks its rig `degraded`, exposed as `intentionallyStopped` in `rig ps --nodes --json`; the `rig ps --nodes` REASON column also now correctly keeps a Codex seat at its usage limit as `unknown` / `usage_limit` instead of `running` while its screen redraws.

— Named flags, fields and migration, evolved coherently across two releases. `v0.6.12` `v0.6.11`

---

09 **Skill storage and plugin lifecycle management** IMPROVED 78

`rig skill loadout --seat <seat>` shows a seat's team skills, now stored in each seat's own storage instead of the project's `.claude/skills` or `.agents/skills`. Startup replaces OpenRig's managed plugin and skill files each release (`openrig-core` moved 0.1.9→0.1.13→0.1.14→0.1.15→0.1.18 across this window); edited copies are backed up under `$OPENRIG_HOME/backups/managed-refresh/` and must be manually reapplied. Claude seats start with `CLAUDE_CODE_PLUGIN_DIRS` pointing at OpenRig's copy and require Claude Code 2.1.280 or newer to pick up hooks and team skills. Claude activity-relay and context-collector hooks in `.claude/settings.local.json` now call the daemon's Node by absolute path instead of bare `node`, and 0.7.1 stops rewriting a worktree's `defaultMode` in that file to `acceptEdits` (already-changed files are not repaired). Seats added to a running rig now receive its culture and startup files, Codex seats selecting `openrig-core` get refocus and skills installed automatically with hook trust granted after a file check, and the refocus hook may now take up to 60 seconds (was 5).

— Many named files and keys, but mostly automatic, little direct action. v0.7.1 v0.7.0 v0.6.13 v0.6.12

10 **Per-seat reasoning effort controls** NEW 78

`rig seat set-effort` and `rig seat effort` support Pi seats, sending the configured `effort` as `--thinking <level>` on every managed launch and resume, with the report showing requested, saved, sent, and recorded values; effort is persisted via `nodes.effort_choice` (migration `104`), and the TUI agent table now shows each seat's configured effort and when its own build version differs from the daemon's.

— Named commands, flag and migration; concrete report fields listed. v0.7.0 v0.6.13 v0.6.12

11 **Work candidate limits and refocus work packets** IMPROVED 78

`OPENRIG_REFOCUS_WORK_PACKET=1` (off by default) enables bounded refocus work packets — one-line summaries ~160 chars with `rig queue show <id>` pointers — initially capped at 20 held and 10 next rows with section bodies ~1,500 chars; `rig queue whoami --work-candidates` now returns up to 50 held and 50 next rows with `heldTruncated` and `nextTruncated` fields (`heldAndNextTruncated` if either was cut), and with the env var on, the refocus hook's work packet also lists held and next work separately, up to 50 each.

— Named env var, command and exact numeric limits across two releases. v0.6.13 v0.6.11

12 **Environment variables for seats and launch specs** NEW 74

Managed Claude seats launched with a native permission mode now receive `EDITOR`, `VISUAL`, `DISPLAY`, `XAUTHORITY`, and an absolute `WAYLAND_DISPLAY`, enabling editor integration (Ctrl+G) and image paste; separately, `rig up --env` now works across

every launch shape so environment variables reach all launch paths, backed by new `rigs.env_json` and `nodes.env_json` columns (migration `111`) carrying environment variables in a rig or seat spec, and a member's `cwd` in a rig spec now also accepts `~`, `~/` `-prefixed` paths, and `$VAR` or `${VAR}` expansions from the daemon's environment when the literal path doesn't exist.

— Named env vars and resulting capability, but automatic not user-driven. Named flag, migration and path-expansion syntax given.

v0.6.13 v0.7.0 v0.6.12 v0.6.11

---

13 **Queue verify, nudge and closure reporting changes** IMPROVED 74

`rig queue create --verify` now reports `outcome: 'claimed'` for a claimed row whose wake wasn't confirmed (instead of `indeterminate`) and reports an agent seat's wake result and any claim as soon as one is recorded instead of waiting 30 seconds; a new `lastNudgeWireResult` field preserves the original wire value when `lastNudgeResult` is overridden to `claimed`. A `done` close can no longer use `--closure-reason blocked_on` — use `rig queue block <id> --on <child>` instead.

— Named fields and commands, clear before/after for each change. v0.7.0 v0.6.11 v0.6.10

---

14 **rig reconcile-session pane repair** NEW 74

`rig reconcile-session <seat> --repair-missing-pane --no-launch` repairs agents with no recorded pane in place, preserving the seat's session, resume token, and history without adopting anything.

— Exact command and flags with preserved-state guarantees. v0.6.14

---

15 **Managed compaction controls and timing** IMPROVED 73

`rig compact --cancel` stops a managed compaction that already sent `/compact` without requiring a daemon restart; `policies.claude_compaction.preparation_root` sets where restore maps are written; the manual preparation timeout increased from 120s to 180s and the request timeout from 180s to 240s.

— Named config key and command with concrete timeout values. v0.7.1 v0.7.0 v0.6.14

---

16 **OMP transient failure retry** IMPROVED 72

OMP seats that ended their turn with a typed transient provider failure (status 408, 429, 500, 502, 503, or 504/529) can now accept an ordinary `rig send` retry instead of staying in needs-input.

— Exact status codes named; command to use given. v0.6.13

---

17 **Usage and cost tracking persistence** NEW 72

`provider.usage_windows` (or `OPENRIG_PROVIDER_USAGE_WINDOWS`) makes `rig provider status --refresh` read a Codex login's usage windows, with a new `provider_account_readings` table (migration `117`) persisting each resolved Codex account's usage readings across daemon restarts when the setting is on; separately, two new `usage_samples` columns (migration `105`) capture Claude's client-side cost estimate for per-seat cost reporting, and new `pi_usage_entries` (migration `107`) and `pi_usage_charges` (migration `108`) tables track per-interaction usage entries and charges for Pi seats, controlled by new `usage.pi_cost_facts` and `usage.pi_openrouter_charge` settings.

— Named config key, env var and command across two releases. Named migrations and settings but no user-facing report surface shown.

v0.6.13 v0.6.11 v0.6.12

---

18 **Safer rig seat launch --fresh --stop** IMPROVED 72

`rig seat launch --fresh --stop` now refuses and exits before stopping anything when a seat has a still-live older session, instead of stopping the agent first; a new `--plan` flag previews a fresh launch without changing anything.

— Named flags with clear before/after safety behaviour. v0.6.11

---

19 **Terminal environment isolation for new agents** BREAKING 72

New agents now start from a clean environment rather than inheriting tmux's global environment; the `terminal.inherit_tmux_environment` config key restores the old behaviour, and values can instead be set via a seat's `env` or `rig up --env KEY=VALUE`.

— Named config key and migration paths for a behavioural change. v0.7.0

---

20 **Standardized error exit codes and HTTP statuses** BREAKING 71

`rig compact` now exits 1 (was 2) and answers HTTP 409 (was 502) when a draft-aware seat has no recorded pane; a transport write that throws without a code makes `rig send` exit 1 with HTTP 409 instead of exit 2 with HTTP 500; `rig queue update` refusals you can fix now answer HTTP 400 and exit 1, not HTTP 500 and exit 2; `rig ask` exits 1 for an unknown or ambiguous rig instead of 0.

— Exact before/after codes for four commands; no migration steps. v0.6.12

---

21 **rig setup provider selection and tmux history default** IMPROVED 70

`--providers` flag on `rig setup` (accepts `claude`, `codex`, `both`, or a comma list) installs and checks only the chosen harnesses, with unchosen providers reading

`skipped` instead of `failed`; `rig setup` also now sets tmux history to 10,000 rows, down from 50,000.

— Named flag with accepted values and a concrete numeric default change. v0.7.0 v0.6.10

22 **rig capture --lines behavior change** BREAKING 70

`rig capture --lines N` now shows the last N lines rather than N history lines plus the whole pane, adding `requestedLines` and `omittedLines` JSON fields; `--history-plus-pane` restores the old behaviour.

— Named flag and fields with clear before/after semantics. v0.6.10

23 **Restore and launch state handling changes** BREAKING 69

After `rig down`, `rig ps` and the TUI now report a rig as `stopped` instead of recoverable. An ordinary terminal seat relaunches fresh on restore instead of stopping at `awaiting-decision`, but restore now stops a seat whose current runtime differs from the one its snapshot started with (`--fresh` does not bypass this check), and `rig launch` of a seat with a recorded resume token from an earlier session now exits 1 and stops at `awaiting-decision` instead of starting fresh. `rig whoami --full --json` also now shows a Codex seat's `runtimeContext.threadId` when its stored conversation ID was verified resumable.

— Several before/after state changes named, but no remediation steps. v0.6.12 v0.6.11

24 **Model observation display** IMPROVED 68

`rig ps --nodes`, `rig roster show`, and `rig roster find` now display the model each seat was last actually observed using, alongside its configured model, with source and timestamp.

— Named commands and fields, no example output shown. v0.6.13

25 **Bundle create refuses harness login files** BREAKING 68

`rig bundle create` now refuses to include harness login files (`.claude/.credentials.json`, `.codex/auth.json`, `.pi/agent/auth.json`); existing archives holding one fail `verify`, `unpack`, and `install`.

— Named files and affected commands, clear before/after behaviour. v0.6.12

26 **Rig launch and naming by name** IMPROVED 64

`rig up <spec> --name <rig>` runs one spec as several named rigs simultaneously; `rig launch` now supports launching a rig by its name instead of only its ID; `rig add` likewise now accepts a rig's name as well as its ID.

— Three named commands grouped as one naming-by-name convenience. v0.7.1 v0.6.12

v0.6.10

27 **Guidance file restoration for rig up --existing** IMPROVED 64

`rig up --existing` now puts back the OpenRig guidance (`CLAUDE.md` , `CLAUDE.local.md` , or `AGENTS.md`) that `rig down` removed before a resumed seat starts, and now exits 1 and reports `partially_restored` when a seat is still missing part of its guidance instead of reporting `fully_restored` .

— Named files and report states, clear mechanism. v0.6.10

28 **rig scope approve workspace stamping** BREAKING 64

`rig scope <tier> approve --workspace <path>` now stamps the named work tree and refuses if that workspace has no `missions/` folder, instead of silently stamping the daemon's own work tree.

— Named command, flag and required folder, clear refusal condition. v0.6.9

29 **NixOS Claude seat recognition** IMPROVED 64

Claude Code seats installed via nixpkgs whose `claude` binary wraps `.claude-unwrapped` are now recognised as Claude and report `running` instead of `attention_required` ; clearing a stale attention marker requires `rig seat clear-attention` .

— Specific platform fix with named recovery command. v0.6.9

30 **Local daemon guard and health endpoint field** NEW 62

`rig config set transport.local_user_guard true` is an optional Mac-only guard making the CLI refuse a local daemon running under a different account; `/healthz` gained an `euid` field alongside the existing `pid` field in the bundled openrig-core plugin (0.1.15).

— Named config key and endpoint field, narrow platform scope. v0.6.14

31 **MCP server over HTTP** NEW 60

`rig mcp serve --http` serves OpenRig's 34-tool MCP server over HTTP so assistants on other machines can connect, with each caller acting as its own seat.

— Command named; tool count noted but internals unexplained. v0.7.1

THINNER COVERAGE BELOW

32 **Rig name dot validation** BREAKING 59

A rig name containing a dot is now refused by `rig up` , adding a member, first-start retry, bundle install, expand, and the preflight preview; existing rigs with a dot in their

name get the same refusal when adding a member or retrying a first start.

— Lists affected surfaces but is a validation rule, not a workflow. v0.6.13

33 **Chatroom command improvements** IMPROVED 59

`rig chatroom history` now returns the newest messages when neither `--after` nor `--since` is given (still listed oldest-first), instead of returning a long room's oldest messages; `rig chatroom wait` now retries through poll timeouts, connection refusals, and reset or closed connections until `--timeout`, rather than ending at the first failed poll.

— Two named subcommands with clear before/after, limited depth. v0.6.10 v0.6.9

34 **New configuration settings and provider.subscription_moves removal**

DEPRECATED 57

0.6.12 added `output.whoami_peers` (default `8`), `bundle.services_enabled`, `provider.subscription_moves`, and `transport.verify_input_box` as new settings; 0.7.0 then removed the `rig provider moves` command and the `provider.subscription_moves` setting — report a subscription move as a queue item instead.

— Named settings and removal, but most lack behavioural description. v0.7.0 v0.6.12

35 **rig ask --wake runtime refusal** BREAKING 57

`rig ask --wake` now refuses a runtime it cannot wake (Pi, custom, or unrecorded) instead of silently attempting a Claude wake; the CLI and daemon must be upgraded together to catch every case.

— Named flag and upgrade requirement, clear before/after. v0.6.14

36 **Watchdog update and context thresholds** IMPROVED 54

`rig watchdog update` retunes a watchdog job in place; the context watch now accepts token or percent thresholds.

— Command and threshold types named, no example values given. v0.7.1

37 **ACP agent dispatch** NEW 53

`rig acp run <row>` hands a row's task to an ACP agent such as Hermes or OpenClaw.

— Single command named with no further mechanism given. v0.7.1

38 **Diagnostic output improvements for queue and ps commands**

IMPROVED 51

`rig queue show` gains `--json` output explaining why a wake didn't land; `rig ps --rig` and `rig ps --host --rig` now report unknown rigs and filter correctly.

— Named flags but thin description of the actual output. v0.7.0

39 **Live edge editing** NEW 50

`rig edge add` and `rig edge remove` change one edge on a running rig without taking it down.

— Commands named but behaviour described in one line. v0.7.1

40 **Kernel recovery and restore** IMPROVED 50

The daemon now automatically restores its kernel after a reboot when all seats are detached; a kernel that recovers from a failed first start now reports ready in `rig status`, with a separate line naming the earlier failure, instead of remaining stuck at `bootstrap_failed`.

— Behaviour described but mostly automatic, little reader action. v0.7.0 v0.6.9

41 **rig ps/tui inventory read optimization** IMPROVED 40

The inventory read behind `rig ps` and `rig tui` refreshes now uses an existing index to find launch fences instead of scanning the whole event log.

— Internal performance change with no user-facing lever. v0.6.14

42 **Quickstart documentation** NEW 32

Adds `docs/reference/quickstart.md`, a getting-started page explaining how OpenRig's pieces fit together and what it changes on the machine.

— Just a doc path named, no content summarized. v0.6.14

└──▷ BREAKING ON UPGRADE

! Eight database migrations apply on first daemon start after upgrade — run `rig daemon stop && rig daemon start` before use.

! Startup replaces OpenRig's own plugin and skill files (`openrig-core` goes from 0.1.17 to 0.1.18); edited copies are backed up under `$OPENRIG_HOME/backups/managed-refresh/` and must be manually reapplied.

! Claude seats require Claude Code 2.1.280 or newer to see their team skills.

! Downgrading to 0.7.0 while a `slack-subscription` is set causes the 0.7.0 people registry to fail loading; use `rig gateway human set <id> slack-subscription unset` before downgrading.

! Any worktree whose `defaultMode` in `.claude/settings.local.json` was rewritten to `acceptEdits` by an earlier version must be manually restored — 0.7.1 stops writing it but does not repair already-changed files.

! New agents no longer inherit tmux's global environment (`tmux set-environment -g …`); put values in the seat's `env`, pass `rig up --env KEY=VALUE`, or set `rig config set`

`terminal.inherit_tmux_environment true` to restore the old behaviour.

! `rig provider moves` command and `provider.subscription_moves` setting are removed; report a subscription move as a queue item instead.

! `rig slack verify` no longer accepts `--reason` or `--actor` flags.

! A `done` close can no longer use `--closure-reason blocked_on`; use `rig queue block <id> --on <child>` instead.

! `rig setup` now sets tmux history to 10,000 rows, down from 50,000.

! If you address queue rows to another machine by its own id (`member@rig@<self-id>`), run `rig host doctor <alias>` once per host after upgrading.

! Startup replaces OpenRig's own plugin and skill files with 0.7.0 copies, backing up originals under `$OPENRIG_HOME/backups/managed-refresh/`; any local edits to those files must be reapplied manually.

! `rig ask --wake` now refuses a runtime it cannot wake (Pi, custom, or unrecorded) instead of silently attempting a Claude wake — the CLI and daemon must be upgraded together to catch every case.

! A manual `rig compact` preparation timeout increased from 120 s to 180 s, and the request timeout from 180 s to 240 s, meaning compaction can take longer to respond.

! A rig name containing a dot is now refused by `rig up`, adding a member, first-start retry, bundle install, expand, and the preflight preview — existing rigs with a dot in their name receive the same refusal when adding a member or retrying a first start.

! Restart the daemon after upgrading the CLI: run `rig daemon stop` then `rig daemon start` to pick up the new build.

! The bundled openrig-core plugin updates from 0.1.13 to 0.1.14; Claude activity-relay and context-collector hooks in `.claude/settings.local.json` are rewritten to call the daemon's Node by absolute path — existing bare-`node` entries are refreshed automatically.

! The daemon must be restarted after upgrading the CLI (`rig daemon stop` then `rig daemon start`); twelve database migrations run on first start.

! `rig start --last` no longer restores a rig stopped with `rig down` or `rig seat stop` — it leaves it down and prints `rig start --rigs <rig>` to bring it back.

! After `rig down`, `rig ps` and the TUI report the rig as `stopped` instead of recoverable.

! An ordinary terminal seat now relaunches fresh on restore instead of stopping at `awaiting-decision`.

! Restore stops a seat whose current runtime differs from the one its snapshot started with; `--fresh` does not bypass this check.

! `rig bundle create` refuses harness login files (`.claude/.credentials.json`, `.codex/auth.json`, `.pi/agent/auth.json`); existing archives holding one fail `verify`, `unpack`, and `install`.

! `rig compact` exits 1 (was 2) and answers HTTP 409 (was 502) when a draft-aware seat has no recorded pane.

! Over the shared transport, a write that throws without a code makes `rig send` exit 1 with HTTP 409 instead of exit 2 with HTTP 500.

- ! `rig queue update` refusals you can fix now answer HTTP 400 and exit 1, not HTTP 500 and exit 2.
- ! `rig ask` exits 1 for an unknown or ambiguous rig instead of 0.
- ! The `fable` model alias now names `claude-fable-5-1`; a seat pinned to `fable` but running `claude-fable-5` will raise a divergence notice.
- ! Migration `109` makes the mode store's unique index per category — any duplicate rows are kept but the old unique constraint is dropped.
- ! Migration `115` rebuilds the Slack thread map keyed by channel and thread timestamp.
- ! The bundled `openrig-core` plugin goes from 0.1.9 to 0.1.13; Claude seats now start with `CLAUDE_CODE_PLUGIN_DIRS` pointing at OpenRig's copy and require Claude Code 2.1.280 or later to pick up the hooks.
- ! The refocus hook may now take up to 60 seconds (was 5), so a hanging refocus can hold up a prompt for up to a minute.
- ! `rig launch` of a seat that has a recorded resume token from an earlier session now exits 1 and stops at `awaiting-decision` instead of starting the seat fresh.
- ! `rig queue create --verify` now reports `outcome: 'claimed'` for a claimed row whose wake wasn't confirmed, instead of `indeterminate`.
- ! `rig send --wait-for-idle N` now treats N as a deadline from the start of the wait rather than from lease acquisition, changing timing semantics.
- ! `rig seat launch --fresh --stop` now refuses and exits before stopping anything when a seat has a still-live older session; previously it stopped the agent first.
- ! The daemon must be restarted after upgrading the CLI (`rig daemon stop` then `rig daemon start`); the running daemon keeps its old build until restarted.
- ! Migration `103` runs automatically on first daemon start and adds the `intentionally_stopped` column to `sessions`; there are no migrations `100` or `102`.
- ! `rig capture --lines N` now shows the last N lines (not N history lines plus the whole pane); add `--history-plus-pane` to restore the old behaviour. In `--json` output, `lines` is now the count returned, and `requestedLines` and `omittedLines` are new fields.
- ! Compact `rig queue list` rows (and `overdue` and `undelivered`) now omit absent fields entirely instead of returning `null` or `""`, and include a new `fieldsElided` field naming what was dropped.
- ! `rig launch <rigId> <seat>` on a seat whose agent exited to a shell now returns `attention_required` and exits 1, instead of reporting the seat already running.
- ! `rig up <rig> --existing` now exits 1 and reports `partially_restored` when a seat is still missing part of its guidance, instead of reporting `fully_restored`.
- ! `lastNudgeResult` now reads `claimed` on a claimed row whose wake was not confirmed; the original wire value is moved to the new `lastNudgeWireResult` field.
- ! `rig chatroom history` now returns the newest messages when neither `--after` nor `--since` is given (still listed oldest-first); previously a long room returned its oldest messages.
- ! `rig scope <tier> approve --workspace <path>` now refuses and exits if the named workspace has no `missions/` folder, rather than silently stamping the daemon's own work tree — scripts that relied on the fallback behaviour will now fail.

! `rig send <session> --host <id>` to an http-registered host now exits 1 with `Delivery UNCONFIRMED` when the outcome cannot be confirmed (e.g. connection drops after the request went out), instead of reporting success or a simple timeout failure — callers must handle the new `failedStep: 'remote-outcome-unknown'` exit code.

WAS THIS USEFUL?  

wrongstack

Feedback ⓘ v1.2.0 NOTES ↗ RANK

An AI coding agent that reads your code, edits files, runs commands, and reasons through bugs — across a terminal REPL, a full-screen TUI, and a browser UI, while you keep your hand on every permission.

> WrongStack's biggest additions this window are the VibeUI coding workspace (`wstack vibeui`) with its own file shelf and Cozy default view, WrongStack ToolFlow for scripted tool orchestration, persistent scheduled GitHub automation, and SAGE memory health diagnostics (`wstack sage doctor`). Smaller releases added durable MCP tasks, ACP Claude Code session leadership, dead-code and security-AST scanning tools, OAuth sign-in for four new model providers, and finer-grained effort and confirmation controls across CLI, TUI and WebUI.

└──▷ **WHAT SHIPPED · 38 FEATURES** most completely described first ? what's the number?

01 **SAGE memory consistency and health** NEW 91

Adds SAGE verify-on-mutation, so edits re-check the memories anchored to changed files, plus `wstack sage doctor`, `/memory doctor`, and a WebUI SAGE health panel reporting on storage integrity, expired memories, graph hubs, dangling edges, audiences, and review backlog, with idempotent repairs that never hard-delete memories; idle SAGE hygiene also runs every six hours via the project daemon, with graph traversal dampening for oversized hubs and stronger ranking penalties for repeatedly injected but unused memories.

Inspect SAGE memory health in a project — surface dangling edges, expired memories, and review backlog before a long autonomous run.

```
$ wstack sage doctor
```

— Explains mechanism but no command or config to trigger it. Names commands, panel, and six-hour hygiene cadence with example.

v1.1.15 v1.1.9

02 **VibeUI coding workspace** NEW 88

WrongStack adds `wstack vibeui` (also `--vibeui`, serving on port 3476), a focused coding workspace between SimpleUI and the full WebUI, with queued chat (Queue / Steer / BTW), provider and account setup, files and diffs, live agents, quotas, a SAGE memory relationship graph, and next-step suggestions that auto-submit under YOLO + Auto. It later gains a file shelf and explorers — pin up to eight project files, revisit five recent files, filter by type, switch card/tree and single/split views, follow folder breadcrumbs, and download a previewed copy — plus a Cozy VibeUI workspace as the first-visit default with Just chat and Studio views, visual starters, and a quick tour.

Launch the VibeUI focused coding workspace to get a lighter interface between SimpleUI and the full WebUI, with queued chat and SAGE memory graph.

```
$ wstack vibeui
```

— Names exact command, port, and workspace features across releases. v1.2.0 v1.1.7

03 **WrongForge queue timeout config** NEW 81

Adds `providers.wrongforge.queueTimeoutMs` config key to bound time spent waiting for WrongForge capacity (default 120 seconds), separately from execution time.

Cap how long WrongForge jobs wait for available capacity before failing, so stalled queue slots don't silently block a run.

```yaml
providers:
  wrongforge:
    queueTimeoutMs: 30000
```

— Names exact config key and default, with a runnable example. v1.2.0

04 **ACP Claude Code session leadership** NEW 78

Enables any installed ACP agent such as Claude Code to act as a session leader via `/model acp/claude-code`, adds delegated Claude workers so Claude can be dispatched as a subordinate worker agent, and removes the Claude sign-in requirement.

Switch the current session to use an installed ACP agent (Claude Code) as the leader model.

```
$ /model acp/claude-code
```

— Names the exact slash command and lists all three changes. v1.1.3

05 **Hosted installer scripts** NEW 73

Adds installer scripts served at `wrongstack.com/install.sh` (macOS/Linux) and `wrongstack.com/install.ps1` (Windows) alongside GitHub release assets.

Install WrongStack on macOS or Linux in one step using the new hosted install script.

```
$ curl -fsSL https://wrongstack.com/install.sh | sh
```

— Gives exact URLs and a runnable install command. v1.0.31

06 **New codebase built-in tools** NEW 70

Adds the `codebase-read-symbol` built-in tool for reading individual symbols from the codebase index, plus a dead-code workflow (`docs/tools/dead-code.md`) with `dead-code-scan` and `dead-code-fix` built-in tools for detecting and removing unused code.

— Names the exact tool but no invocation example. Names both tools and the workflow doc precisely.

v1.0.37 v1.0.34

07 **Confirmable tools prompt on default tier** IMPROVED 68

Confirmable tools (`tool_use` , `mcp_use`) behind the token-saving tier now prompt for confirmation like direct calls, enabling `pwsh` , `exec` , `fetch` , `git` , and MCP tools to work when YOLO is off on the default tier.

— Names every affected tool and the tier condition precisely. v1.1.6

08 **OAuth sign-in for four new providers** NEW 68

Adds OAuth sign-in for xAI/Grok, Kimi Code, Meta and ChatGPT plan API across CLI, TUI and WebUI, with account-scoped model discovery and coordinated credential refresh.

— Names all four providers and the surfaces supporting sign-in. v1.0.31

09 **AST security rule profiles for 12 languages** IMPROVED 66

Expands `security-ast-scan` with rule profiles for Python, Go, Rust, Java, Kotlin, Scala, C#, PHP, Ruby, Swift, C/C++, and shell.

— Names the exact tool and all twelve covered languages. v1.1.9

10 **Automation workflow enhancements** NEW 64

Adds Docker environment profiles for automation runs with a live environment panel in WebUI and SimpleUI, independent verification checks, and reviewed draft pull-request publication, alongside persistent automation where scheduled and signed GitHub jobs run in independent Docker source snapshots with CLI, WebUI and SimpleUI controls, versioned templates, portable definitions, structured results, and reviewable patches.

— Groups two thin automation additions naming their surfaces. Describes scope and controls without naming a specific command.

v1.1.13 v1.0.31

11 **WrongStack ToolFlow scripted orchestration** NEW 64

Adds WrongStack ToolFlow: compose repeated or dependent tool calls in sandboxed JavaScript, with permission checks and session records preserved per call; WebUI, TUI and SimpleUI display run purpose, call counts, failure counts and measured output bytes.

— Explains mechanism and metrics but no invocation syntax. v1.0.31

12 **Durable MCP tasks with recovery** NEW 62

Adds durable MCP tasks with multi-round request support, a recovery journal, and a WebUI tasks panel; connections now probe the modern MCP protocol first with an explicit legacy mode fallback.

— Describes mechanism and UI surface but no exact command. v1.1.13

13 **Explicit continuation via /nextsteps** NEW 62

Adds explicit continuation via the `/nextsteps` slash command (`docs/slash/nextsteps.md`), giving users direct control over what the agent pursues next.

— Names the exact command and its doc reference. v1.0.34

THINNER COVERAGE BELOW

14 **Text selection mode in TUI** NEW 58

Adds a Text selection mode in the TUI alongside Block mode, copying only the selected visible text; switchable from Settings or the composer footer.

— Names exact switch locations in the UI. v1.1.7

15 **/provider-quota TUI browser** NEW 58

Adds a `/provider-quota` TUI browser with a structured two-pane layout for inspecting provider quota details.

— Names the exact slash command and its layout. v1.0.31

16 **Session Story and model dashboards** NEW 58

Adds Session Story and model dashboards: a selected-tab timeline, activity charts, replay, team branches and recorded tool/file/memory/mail evidence make session and subagent work inspectable, with model comparisons of reliability, latency, spend and work signals.

— Lists dashboard elements but no navigation path given. v1.0.31

17 **Live model-output activity view** NEW 57

Adds a live model-output activity view across CLI, TUI, SimpleUI and WebUI showing estimated live token counts and speed, elapsed time, and waiting/thinking/receiving/tool-running phase indicators.

— Describes surfaces and signals but no command to invoke. v1.2.0

18 **WrongForge session tool context forwarding** NEW 57

Sessions started on the `wrongforge` provider now forward turns to WrongForge with the session's tools attached, and tool calls appear as tool cards in the UI.

— Names the provider and UI effect but no config path. v1.1.6

19 **Project goals support** NEW 57

Adds `wstack` project goals support via `docs/architecture/project-goals.md`, letting agents loop toward a locked, verifiable contract.

— Names the exact doc path but mechanism stays high-level. v1.0.34

20 **Expanded project environment detection** IMPROVED 55

Extends project environment detection to include Deno, Gradle Kotlin, CMake, Make, Swift, Dart/Flutter and .NET markers, and adds the `language` toolchain check to the tool coach.

— Lists new ecosystems detected but no usage path given. v1.2.0

21 **Offline routing and quality experiments** NEW 53

Adds offline routing and quality experiments: benchmark and transcript commands compare deterministic grades, recorded usage, known cost and advisory review evidence without changing live routing.

— Describes comparison data but not exact command names. v1.0.31

22 **ACP SDK 2 stable/unstable split** IMPROVED 52

Updates ACP SDK 2 to keep stable v1 APIs separate from the `sdk/unstable` entry point for experimental methods.

— Names the exact unstable entry point but limited mechanism detail. v1.1.13

23 **Plugin tool-registration boundaries** BREAKING 52

Plugins are now restricted from modifying tools they did not register, and only official plugins may register tools under the `core` namespace.

— Names the `core` namespace restriction but no migration steps. v1.1.6

24 **Code Assist runs and fleet mail** NEW 52

Adds Code Assist isolated runs and fleet mail: the WebUI starts assistance runs for files and symbols and composes or replies to typed fleet mailbox messages.

— Names the UI entry points but no exact command or endpoint. v1.0.31

25 **Effort routing for leader and worker agents** NEW 51

Adds phase-aware leader effort, adjusting how much effort the leader agent expends based on the current phase of a task, plus per-worker effort routing enabling different

effort levels to be directed to individual worker agents.

– Explains both mechanisms but no config surface to set levels. v1.0.36

26 **Expanded companion working budgets** IMPROVED 48

Expands companion working budgets so Explore, Memory and Skill companions inherit practical role budgets instead of single-lookup limits; Explore can now use bounded repository orientation surveys.

– Explains behaviour change but no concrete limits or config. v1.2.0

27 **Request residency control** NEW 48

Adds request residency control via the `/context` slash command, letting users govern where requests are processed.

– Names the slash command but not the residency options. v1.1.0

28 **YOLO+ autonomous execution tier** NEW 48

Adds YOLO+ mode (`docs/slash/yolo.md`), an expanded autonomous execution tier beyond the existing YOLO setting.

– Names the mode and doc but not its exact behaviour difference. v1.0.34

29 **Browser interaction capabilities** NEW 43

Adds browser setup capability and browser element selection, letting you pick an element in the live browser view and add it to the prompt.

– Names a clear UI action but no further detail. Bare one-line mention with no detail.

v1.1.13 v1.0.37

30 **Reworked WebUI workbench** IMPROVED 43

Adds a reworked WebUI workbench with locally served provider logos and Tab navigation between the desktop shell and its WebUI.

– Describes change but limited mechanism beyond navigation. v1.1.7

31 **Session-owned artifacts presentation** NEW 41

Adds session-owned artifacts: text, images, diffs and owned live browsers can be presented across WebUI and SimpleUI.

– Lists artifact types but no interaction detail given. v1.0.31

32 **Browser auto-submit under Auto + YOLO** IMPROVED 38

Enables browser interfaces to submit the next-step suggestion automatically under Auto + YOLO mode.

– Short behaviour note without further configuration detail. v1.1.9

33 **Capability-gated media routing** NEW 35

Routes image and PDF parts only to models that declare support for them, preventing capability mismatches.

— States behaviour only; no named flag or setting. v1.1.15

34 **Last-request cache readouts** NEW 33

Adds last-request cache readouts so practitioners can inspect what was cached on the most recent request.

— States purpose only, no command or UI path given. v1.1.0

35 **Native cloud profiles and preflight diagnostics** NEW 27

Adds native cloud profiles and read-only preflight diagnostics for provider operations.

— Bare mention without scope or named provider list. v1.0.31

36 **Leaner tool-error context** IMPROVED 18

Delivers leaner tool-error context, reducing noise in tool failure output.

— Bare description with no mechanism or scope. v1.1.0

37 **Skill Companion** NEW 18

Adds the Skill Companion for skill suggestions.

— Names the companion but no mechanism or scope. v1.0.37

38 **Audit-driven runtime hardening** IMPROVED 16

Introduces audit-driven hardening across the agent runtime.

— Bare mention with no specifics on what hardened. v1.0.36

WAS THIS USEFUL? 👍 👎

keploy

Feedback ⓘ v3.6.162 NOTES ↗ RANK

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

> Keploy's twelve releases centered on a new `KEPLOY_PROVEN` strict-comparison mode, capture-loss tracking for incomplete recordings, a `-t` flag for targeted test-set replay, Unix/UDP protocol capture via eBPF, and a string of smaller reliability, tracing and determinism improvements across its recording and proven-replay engine.

└──▷ **WHAT SHIPPED · 18 FEATURES** most completely described first ? what's the number?

01 KEPLOY_PROVEN strict mode for test comparisons NEW

91

The `KEPLOY_PROVEN` environment variable, introduced as a staging switch carried through to the agent, now gates a strict response comparison mode (O-SF): a test set with unread misses is left unjudged rather than silently passing (O-1), a test fails when a miss occurs or a mock is unconsumed at the end of a run (O-2), and skipped tests now report the reason they were skipped (O-3). Example: `KEPLOY_PROVEN=true keploy test -c "python main.py" --delay 10`.

Run your test suite in strict proven mode so any unmatched response, unread miss, or unconsumed mock causes an explicit failure instead of a silent pass.

```
$ KEPLOY_PROVEN=true keploy test -c "python main.py" --delay 10
```

— Exact env var, named behaviors, and a runnable command v3.6.118 v3.6.117

02 Test set selection via -t flag NEW

75

Adds the `-t` flag to specify which recorded test sets to run during replay, e.g. `keploy test -c "python main.py" --delay 10 -t <test-set-name>`.

Run only a specific subset of recorded test sets during CI replay to speed up targeted validation.

```
$ keploy test -c "python main.py" --delay 10 -t <test-set-name>
```

— Exact flag and runnable command given v3.6.129

03 Variable tracing through dependency calls (Proven A2) NEW

60

Proven A2 tracks a test's variables through its dependency calls on the agent side, and binds request variables while checking captures against dependency calls on the CLI side in `keploy test`, completing end-to-end variable tracing across the full call graph.

— Agent and CLI sides explained, no example run shown v3.6.123

THINNER COVERAGE BELOW

04 **Capture-loss tracking for replay** NEW 58

Replay introduces a `capture-loss` kind for app processes that the replay engine could not mark, tainting every test set that contains that process, and adds capture-loss scope tracking to surface gaps in recorded traffic coverage.

— Names the capture-loss kind but no inspection command v3.6.161 v3.6.129

05 **Container-key naming for docker-mode agents** NEW 55

Docker-mode agents now support container keys, sourced from a list the CLI pushes, enabling named identification of container destinations in Proven.

— Mechanism named, no config key or command given v3.6.123

06 **Mock replay clock determinism** IMPROVED 50

Mock replay's clock now advances to each individual test's recorded begin time and shifts at scope marks, improving determinism in proven-mode test runs and time-based replay.

— Mechanism named, no config or command given v3.6.122 v3.6.116

07 **Unix socket and UDP protocol capture** NEW 50

Adds Unix socket and UDP protocol capture via eBPF network interception, expanding the traffic types Keploy can record.

— Protocols and mechanism named, no config shown v3.6.129

08 **Recording reliability improvements** IMPROVED 48

Recording now batches handling of in-flight calls to improve capture reliability for concurrent requests, uses an event-driven set-end stop mechanism to more accurately determine when a recording session has completed, and surfaces inconsistent-recording warnings when captured traffic may be incomplete or unreliable.

— Three mechanisms named, no flags or commands given v3.6.138

09 **Template captures for test-set variables** NEW 45

Introduces template captures: a test set's variables can now be written directly inside test cases, letting tests reference shared variable values inline.

— Capability named but no syntax or example shown v3.6.117

10 **Proven engine batch K-B6 additions** NEW 43

Adds `mock.real` support, path-shape privacy, OSS lanes on wire v1, and test hygiene improvements to the proven engine (batch K-B6).

— Several surfaces named but no mechanism explained v3.6.126

11 **Local port connection refusal in mock replay** NEW 40

Mock replay now refuses connections to local ports that have nothing listening, preventing silent test failures from unexpected local traffic.

— Behavior described, no flag or config surface named v3.6.116

12 **JUnit test-set verdicts** NEW 40

Adds JUnit set-level verdicts so test-set pass/fail results are reported in JUnit output.

— Names JUnit output but no file path or flag given v3.6.129

13 **K1 agent primitives for K-B30 vehicle agent** NEW 35

Adds K1 learn primitives including one-step test opening, variables, and clock reads to the K-B30 vehicle agent.

— Primitives named but no usage shown v3.6.155

14 **Repeated-call miss tracing to earlier recording** IMPROVED 35

On a miss for a repeated call, Keploy now names the earlier test's identical recording so practitioners can trace which captured interaction matches.

— Diagnostic behavior described, no example given v3.6.162

15 **Body kind classification for recordings** NEW 30

Adds `body kind` classification to distinguish payload types in recorded interactions.

— Bare description, no detail on types or usage v3.6.129

16 **Outside-test consumption tracking** NEW 30

Adds outside-test consumption tracking to detect dependency calls made outside active test boundaries.

— Purpose described only, no mechanism detail v3.6.129

17 **Agent process identification via PID logging** IMPROVED 30

The agent's start line now logs its PID so the process is identifiable by the same name used in its responses.

— Simple logging change, no command shown v3.6.162

18 **Universal Mocker protocol v1 scoping** IMPROVED 25

Scopes protocol v1 for the Universal Mocker's single worker in the proven subsystem.

— Bare description with no behavior detail v3.6.121

UPDATED INTEGRATION MySQL credential redaction

> v3.6.127 NOTES ↗ STABLE
>
> Keploy v3.6.127 introduces MySQL credential redaction so captured recordings no longer expose database passwords.
>
> └──▷ GET THIS VERSION ▾
>
> ```
> $ git clone --branch v3.6.127 https://github.com/keploy/keploy.git
> # already have the repo? check out this version:
> $ git checkout v3.6.127
> ```
>
> Adds MySQL credential redaction to prevent sensitive credentials from being stored in recorded test cases.

WAS THIS USEFUL? 👍 👎

prflow

Feedback ⓘ v2.62.0 NOTES ↗ RANK

Transforms feature requests into review-ready pull requests with implementation, tests, review loop, and docs. Runs locally or on GitHub Actions for complex repos.

> PRFlow added two new top-level commands — `/prflow:challenge` to judge a change against its stated goal and `/prflow:help` for setup diagnostics — while significantly hardening `/prflow:spec`'s decision-making, self-auditing and filing flow across six releases. The window also brought several breaking changes: narrower cloud `gh api` grants, removed or changed flags on internal helper scripts, a moving `claude-code-action@v1` workflow tag, and reworked telemetry configuration.

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

01 Cloud `gh api` grant scoped to repository endpoints

BREAKING

90

Cloud runs' built-in `gh api` grant no longer covers `gh issue create`, `gh issue comment`, `gh pr comment`, `gh pr create`, `env`, or `export` — only the run's own repository endpoints are permitted. Shipped workflows changed as part of this; cloud-tier users must update the `prflow_version` pin, re-run the installer (`DEVFLOW_REF=v2.58.0 bash devflow-install.sh --apply`), and re-vendor any committed vendored copy in the same upgrade.

Upgrade a cloud-tier installation to v2.58.0, including the mandatory workflow update and re-vendor step.

```
$ DEVFLOW_REF=v2.58.0 bash devflow-install.sh --apply
```

— Names every restricted command plus exact upgrade command. v2.58.0

02 Spec session flow and decision-quality refinements

IMPROVED

85

`/prflow:spec` removes its clarification question limit so replying `just create it` no longer halts remaining questions, makes its offer to trigger `/prflow:implement` conditional on `workflows.prflow` being set to `true` in `.prflow/config.json`, re-reads each `Blocked by` issue before filing to verify its closure status has not changed since design and enforces completed draft preparation for audits and non-interactive filing, has background sessions each enter their own worktree before discovery starts, states each instruction-change case's expected result as what the changed instructions decide for it and quotes the deciding fields of any recorded input so the implementing

run can replay them, reduces research redundancy by no longer repeating whole-repository probes and reusing option-check findings so fewer repeat option checks run before the implementation-approach question, ends its turn and resumes on the subagent completion notification on Claude Code's interactive tier instead of blocking, and sharpens decision evidence so paraphrases and unchecked conclusions no longer count as verified, disproved approaches are re-asked immediately, the approach question states what the run read to settle an unconfirmed fact, and it now shows its reasoning before prompting for a decision, including evidence against recommended answers, unconfirmed facts behind recommendations, and a walk of each motivating case and named rule arm.

Gate the `/prflow:implement` trigger offer in spec sessions to projects that have explicitly opted in.

```json
{
  "workflows": {
    "prflow": true
  }
}
```

— Named config key with a working example. Clear mechanism described, no runnable example. Single-line description with no further scope. Mechanism described but nothing directly actionable. Behavior change described, nothing to act on directly. Specific to one tier, no further mechanism given. Rich mechanism across two releases, no runnable example.

v2.62.0 v2.61.0 v2.60.0 v2.58.0

---

03 **Breaking changes to internal helper script CLIs** BREAKING

85

`workpad.py` no longer accepts `--record-audit-inputs` or `reuse-check … audit` — direct helper callers using either argument will break on upgrade. `efficiency-trace.sh` now refuses the `--correct-iter` and `--mode` flags, `normalize-verdicts.py` now requires `--contract 2`, `review-dirty-tree.sh` drops the `view-materialize` subcommand, and `stale-prose-lint.py` and `match-lint-adjudications.py` are deleted entirely. Cloud-tier users must also update the `prflow_version` pin and re-run the installer (`DEVFLOW_REF=v2.57.0 bash devflow-install.sh --apply`) in the same upgrade; mixed versions get no compatibility path.

— Every removed flag, subcommand and script named verbatim. v2.60.0 v2.57.0

---

04 **Telemetry and retrospective storage changes** BREAKING

85

The weekly retrospective now saves learnings to the telemetry branch instead of opening a state pull request, and the installer stages removal of tracked learnings files,

which must be committed. Setting `prflow_review_and_fix.efficiency_telemetry_enabled` to false no longer prevents run files and cloud cost records from being sent to the telemetry branch — `telemetry.enabled` must be set to `false` in `.prflow/config.json` instead.

Keep run files, checklists, and cloud cost records off the telemetry branch entirely for a repository where that data must stay private.

```json
{
  "telemetry": {
    "enabled": false
  }
}
```

— Exact config keys and replacement setting given. v2.57.0

05 **Shipped workflows track moving `claude-code-action@v1` tag** 80

BREAKING

Shipped workflows now use the moving `anthropics/claude-code-action@v1` tag instead of a pinned SHA, allowing the bundled Claude Code version and model aliases to change between reruns of the same commit. Cloud-tier users must update the `prflow_version` pin and re-run the installer fetched at v2.60.0 (`DEVFLOW_REF=v2.60.0 bash devflow-install.sh --apply`) to pick up the new tag.

— Exact migration command given. v2.60.0

06 **`/prflow:help` setup diagnostics command** NEW 80

`/prflow:help` checks your setup and a failed cloud run's log, then returns one of three verdicts — setup problem with the fix, likely PRFlow bug, or undetermined; a likely bug produces a local report to share without posting.

Get a plain-language diagnosis when a cloud run fails or your local setup seems misconfigured — without posting anything publicly.

```
$ /prflow:help
```

— Named command with defined verdicts and runnable invocation. v2.57.0

07 **`DEVFLOW_BASH` environment variable** NEW 75

`DEVFLOW_BASH` explicitly selects which `bash` executable PRFlow's Python helpers use on native Windows, useful when multiple bash binaries (e.g. Git Bash vs. WSL) are present.

Pin a specific bash binary for PRFlow on a Windows machine where multiple bash executables exist (e.g. Git Bash vs. WSL).

```
$ set DEVFLOW_BASH=C:\Windows\System32\bash.exe
```

— Named env var with a runnable example. v2.59.0

08 **`/prflow:challenge` command for goal verification** NEW

70

`/prflow:challenge` judges a change against its stated goal: it evaluates what the change actually alters, checks whether removed or suppressed behavior has dependents, and prompts the user when only they hold a fact that could change the verdict.

— Command and behavior named; no usage example given. v2.62.0

09 **Self-audit and filing preview for spec drafts** NEW 60

`/prflow:spec` now audits each saved draft before display, re-checks it after fixes, and allows unlimited further audit rounds. The filing preview now includes the draft path, audit summary, and notes, shortening the filing question.

— Clear mechanism, no flags or commands to invoke. v2.60.0

10 **Checklist verifier behavior and efficiency** IMPROVED

60

Checklist verifiers no longer run shell commands — they read supplied source snapshots and PR diff, and report INCONCLUSIVE when a claim requires a command or evidence they lack; under `/prflow:implement`, such a criterion no longer forces an extra shadow review. Local reviews also spend less time waiting on checklist verifiers, since a verifier is now instructed to finish any in-progress check before saving its verdict.

— Clear before/after mechanism, no direct command. v2.58.0 v2.57.0

11 **`/prflow:implement` reliability improvements** IMPROVED

60

`/prflow:implement` now checks a criterion about prose content by reading that text; its cleanup pass runs before sweeps and tests, which now cover its edits; a waived-review run finishes with a workpad note; and a clean review no longer risks a false Blocked.

— Several concrete behavior fixes, no example given. v2.57.0

THINNER COVERAGE BELOW

12 **`/prflow:review-and-fix` decline logic** IMPROVED 55

`/prflow:review-and-fix` now requires a listed reason to decline a Suggestion or Minor finding, allows a later fix to re-check an earlier decline, and can list a shadow review's new low-severity findings in its `## Coverage` report.

— Named report section, no runnable command. v2.58.0

13 **Windows shadow review directory symlink fix** IMPROVED

35

On Windows, a review run no longer deletes or moves files through a symbolic link at its shadow review directory.

— Bug fix stated plainly, no reproduction steps. v2.60.0

14 **Helper agents use local checkout for GitHub lookups** IMPROVED

25

Helper agents run GitHub lookups from your local checkout.

— Single-sentence description with no further detail. v2.60.0

┗━▷ BREAKING ON UPGRADE

! `workpad.py` no longer accepts `--record-audit-inputs` or `reuse-check … audit` — direct helper callers using either argument will break on upgrade.

! Cloud-tier users must update the `prflow_version` pin and re-run the installer fetched at v2.60.0 (`DEVFLOW_REF=v2.60.0 bash devflow-install.sh --apply`) to pick up the new `anthropics/claude-code-action@v1` workflow tag.

! Cloud runs' built-in `gh api` grant no longer includes `gh issue create` , `gh issue comment` , `gh pr comment` , `gh pr create` , `env` , or `export` — any workflow relying on these from a cloud run will break.

! Shipped workflows changed; cloud-tier users must update the `prflow_version` pin, re-run `DEVFLOW_REF=v2.58.0 bash devflow-install.sh --apply` , and re-vendor any committed vendored copy in the same upgrade.

! `efficiency-trace.sh` now refuses the `--correct-iter` and `--mode` flags — callers using either will break.

! `normalize-verdicts.py` now requires `--contract 2` — existing invocations without it will break.

! `review-dirty-tree.sh` drops the `view-materialize` subcommand — callers using it will break.

! `stale-prose-lint.py` and `match-lint-adjudications.py` are deleted — any callers of either script will break.

! The weekly retrospective now saves learnings to the telemetry branch instead of opening a state pull request; the installer stages removal of tracked learnings files, which must be committed.

! Setting `prflow_review_and_fix.efficiency_telemetry_enabled` to false no longer prevents run files and cloud cost records from being sent to the telemetry branch — `telemetry.enabled` must be set to false instead.

! Cloud-tier users must update the `prflow_version` pin and re-run the installer at `v2.57.0` (`DEVFLOW_REF=v2.57.0 bash devflow-install.sh --apply`) in the same upgrade; mixed versions get no compatibility path.

WAS THIS USEFUL? 👍 👎

Graphify Labs Graphify

Feedback

announced · Oct 25, 2026 BLOG ↗ RANK

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

> Graphify added Google Drive as a connectable workspace source and extended structural extraction to YAML files, letting Helm charts, Kubernetes manifests, and CI pipelines contribute graph nodes.

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

01 **Structural YAML extraction into the knowledge graph** NEW

85

Graphify now extracts `.yaml` / `.yml` files structurally, producing a file node plus `contains` edges to keys, classified as code so Helm charts, Kubernetes manifests, and CI pipelines contribute nodes to a code-only or `--update` run. Multi-document streams separated by `---` are walked per document, output is bounded at 1 MiB, 500 pairs, and depth 6, and parsing uses `safe_load` so a `!!python/...` tag never executes. The AST cache schema was bumped for this change, so the first build after upgrading will re-extract the entire codebase.

— Names flags, limits, and parser safety mechanism precisely v0.9.95

02 **Google Drive workspace connector** NEW

81

A Google Drive folder (including shared drives) can now be added as a repository and made searchable via Open graph, Wiki, and Ask after the first build. It reads Google Docs, Sheets, and Slides plus PDF, Word, Excel, PowerPoint, CSV, Markdown, text, and image files, and workspace members can search the folder's contents even if they lack direct Drive access to the original file — adding a folder requires checking 'I understand everyone in this workspace can read this folder's contents'. Graphify re-lists the folder every 15 minutes and rebuilds the graph only when the file list changed, backing off up to 6 hours on failed checks; disconnecting revokes read-only Drive access but leaves the already-built graph searchable, and reconnecting the same account leaves existing folders in place.

— Full mechanism and config named; starting point is a UI connect flow

launch-20261025-8b321500

└──▷ BREAKING ON UPGRADE

! The AST cache schema is bumped; the first build after upgrading will re-extract the entire codebase.

WAS THIS USEFUL? 👍 👎

The Open Engine Zeroshot

Feedback ⓘ v10.18.0 NOTES ↗

RANK

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

> Zeroshot rebranded its hosted Cloud target to Platform while keeping legacy Cloud workspaces intact, and shipped two substantial new capabilities: Active Memory for learning from past run history, and native GitHub issue/PR review delivery via dedicated graphs.

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

01 **Native GitHub issue and PR review delivery** NEW 93

Adds separate `issue-review` and `pr-review` graphs for posting structured review reports to GitHub, with durable publication records, retry-safe comment reuse, and reproduction/setup evidence preserved in comments. Supports native GitHub comment delivery and pre-start issue/PR URL intake, and custom graphs can use the same publisher.

Run a structured PR review that posts a durable comment with reproduction and setup evidence directly to the pull request.

```
$ zeroshot run \
  --title "Review PR #42" \
  --template pr-review \
  --input input.json \
  --uniform-runtime-config runtime.json
```

— Named graphs and runnable command demonstrate exact usage v10.17.0

02 **Active Memory for local and Docker runs** NEW 90

Adds the `zeroshot memory init --directory DIR` command and `zeroshot-memory-init` skill to initialize Active Memory from local run history, with repository/graph exclusions and reconciliation against current checkouts. Active Memory is enabled by default for local and direct Docker runs, recalling lessons from past repository mistakes and recurring graph procedures via deterministic hooks and instruction guidance.

Seed Active Memory from an existing repo's run history so future runs avoid known mistakes without manual curation.

```
$ zeroshot memory init --directory /path/to/repo
```

— Exact command and default-on behaviour with mechanism explained v10.17.0

03 **Platform target replaces Cloud target** NEW 85

Adds the built-in `platform` target to fresh registries; sign in with `zeroshot target login platform` and select it with `--target platform`. Adds `kind: "platform"` for hosted workspaces, starting fresh browser drafts and preferences without a state-migration layer. Existing `cloud` target records retain their saved identity and remain usable alongside the new `platform` target.

Authenticate against the new Zeroshot Platform target after upgrading from a registry that previously used `cloud`.

```
$ zeroshot target login platform
```

Run a software-change task explicitly routed through the Platform target instead of the legacy Cloud target.

```
$ zeroshot run \
  --title 'Add JSON status output' \
  --template software-change \
  --input input.json \
  --uniform-runtime-config runtime.json \
  --target platform
```

— Names exact commands, flags and config key with retained compatibility v10.18.0

THINNER COVERAGE BELOW

04 **Quick start and first-run guide documentation updates** IMPROVED 45

Updates `docs/llms.txt` quick start to pass `--delivery none`, document that a valid request prints `{"valid":true}`, and name harness/provider pairs for Claude Code, GitHub Copilot and Pi. Adds a `runtime.json` example to the first-run guide for every supported harness.

— Documentation-only change, no runnable command for readers v10.18.0

WAS THIS USEFUL? 👍 👎

planning-with-files

Feedback ⓘ v3.25.2 NOTES ↗ RANK

Persistent file-based planning for AI coding agents with crash-proof markdown plans, session recovery, and context-rot protection. Works with Claude Code and 60+ agents.

> This window's release cycle centers on plan integrity checks and multi-plan selection across more agent integrations, alongside a stricter attestation status for OpenCode and refined pre-tool reminder behaviour for Pi.

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

01 Plan format validation via check-complete.sh and plan-doctor.sh

NEW 85

Adds a `--validate` flag to `check-complete.sh` (and `-Validate` to its PowerShell equivalent) that rejects plans with no recognized phases or with missing/ambiguous native phase statuses before letting a gated stop proceed. Adds a separate `plan-doctor.sh` diagnostic that reports the same problems without rewriting the plan or changing its attestation.

Validate a plan's phases and statuses before letting a gated stop proceed — catches missing or ambiguous phase statuses early.

```
$ check-complete.sh --validate
```

Inspect a plan for structural problems (missing phases, ambiguous statuses) without modifying the plan or its attestation.

```
$ plan-doctor.sh
```

— Names exact flags and scripts with runnable commands v3.25.2

02 Session-scoped Pi plan reminder suppression with PWF_PRETOOL override

IMPROVED 70

Introduces session-scoped suppression of repeated Pi pre-tool plan views after confirmed turn-start context insertion, with a single refresh queued on visible plan edits and delivery-state invalidation on selection changes, recovery events, and failed or aborted runs. Adds the `PWF_PRETOOL=always` environment variable to restore the previous per-leaf Pi plan injection behaviour when this suppression is unwanted.

— Mechanism detailed and env var override named v3.24.2

03 **Named plan selection across Cursor, Hermes, and DSH integrations**

IMPROVED 65

Cursor bash hooks now consistently select named plans and refuse ambiguous or invalid selection rather than falling back to an unrelated root plan. Hermes adapter 0.3.0 adds a `planning_with_files_select` tool to select or clear a conversation's active plan, including plans in other projects, backed by profile-local records that survive backend restarts; missing/invalid selections fail closed and conflicting explicit directory arguments return an error. DSH `/pwf-status` now preserves the `PLAN_ID` selection notice when multiple plans exist, and still emits the initialization message for projects with no plan.

— Names tool and status surfaces but no runnable example given v3.25.0 v3.25.1 v3.25.2

THINNER COVERAGE BELOW

04 **attestationValid digest-check status for OpenCode** NEW 45

Adds an `attestationValid` status field to OpenCode which requires a digest check against the current plan, distinct from the presence-only `attested` marker.

— Explains distinction but no usage path shown v3.25.1

05 **Skill-lens evaluation framework for planning skill impact** NEW 30

Adds an optional skill-lens evaluation framework that compares planning-file creation with and without the skill, including contributor-reported measurements, checks' limits, and paid-run requirements.

— Describes purpose only, no mechanism or command named v3.25.1

06 **Copilot context injection via additionalContext** IMPROVED 25

Copilot startup and post-tool context now uses the current harness's top-level `additionalContext` output.

— Single bare statement of what changed v3.25.2

NEW INTEGRATION Qoder CLI native plugin

> v3.24.0 NOTES ↗ STABLE
>
> Adds `PWF_PRETOOL=always` to force per-call context injection; suppresses repeated unchanged PreToolUse views in Claude Code with per-session/agent/plan state.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v3.24.0 https://github.com/OthmanAdi/planning-with-files.git
# already have the repo? check out this version:
$ git checkout v3.24.0
```

Adds `PWF_PRETOOL=always` environment variable to request planning context on every matched tool call, overriding suppression of unchanged PreToolUse views.

Suppresses repeated unchanged PreToolUse planning-context injections in Claude Code plugin and standalone skill hooks after successful prompt injection, refreshing only when the view changes; sessions, agents, selected plans, and rendering modes each maintain separate state.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen Code

Feedback ⓘ desktop-v0.26.1 NOTES ↗

RANK

An open-source AI coding agent that lives in your terminal.

> Across two releases this window, Qwen Code deepened its managed-agent and daemon tooling: durable and idle-only context compression for managed sessions, hot-swappable MCP servers and default subagent delegation, a persisted daemon bearer token via `qwen daemon start --save-token`, and a new `/autofix-pr` automation layer for PR workflows.

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

01 **Daemon CLI improvements** NEW 90

Qwen Code adds a single command to start both the daemon and Managed locally together, plus `qwen daemon start --save-token` to persist the bearer token in an owner-only file and `qwen daemon token` to inspect or rotate the stored token.

Start the daemon and persist the bearer token to disk so long-running automation scripts can reconnect without re-authenticating.

```
$ qwen daemon start --save-token
```

Inspect or rotate the stored bearer token after the daemon has been started with `--save-token`.

```
$ qwen daemon token
```

— Command exists but exact syntax not given Exact commands with runnable examples given

desktop-v0.26.1 v0.26.1

02 **max-sessions persistence in settings.json** NEW 75

Qwen Code adds `max-sessions` persistence in `settings.json`, letting the maximum number of concurrent sessions survive daemon restarts.

Persist the maximum number of concurrent sessions across daemon restarts.

```json
{ "max-sessions": 4 }
```

— Exact config key with example value given v0.26.1

03 **PR workflow automation** NEW 70

Qwen Code binds the reviewed PR automatically when a `/review` prompt is admitted, and adds the `/autofix-pr` reaction layer with guarded push, sync, replies, re-runs, and skill support, plus PR watch mode via `feat(verify-pr): add a watch mode backed by the PR probe gate`.

— Named slash command with brief mechanism Named command and watch-mode mechanism but no runnable example

desktop-v0.26.1 v0.26.1

04 **Managed-agent session interface enhancements** NEW 65

Qwen Code adds durable manual context compression for managed-agent sessions, letting users explicitly compress conversation history without losing session continuity and safely resume sessions, plus idle-only manual context compression for Workspace-bound Managed Sessions exposed via the public API and the WebShell context-usage popover (preserving audit history); it also adds session lifecycle UI and reading exports to the managed-agent interface, surfacing session state transitions and exportable session data, with the session browser gaining paginated checkpoint history, improved session feedback, and filtered session title queries.

— Named API and UI surface given, no exact endpoint syntax UI additions named but no exact navigation or flags given

v0.26.1 desktop-v0.26.1

05 **Switchable Linux runner fleet for CI** NEW 60

Qwen Code adds the `LINUX_RUNNER_FLEET` variable and a dispatch option for a switchable Linux runner fleet (hosted/arm64).

— Named environment variable with concrete values v0.26.1

THINNER COVERAGE BELOW

06 **tool.check prompting for Mods** NEW 55

Qwen Code adds `tool.check` prompting for Mods, routing a Mod's own tool permission ask to the user interactively.

— Named flag but no runnable example given desktop-v0.26.1

07 **MCP server hot-management without session recreation** NEW

45

Qwen Code enables MCP server management — adding, removing, and reconfiguring servers — without recreating existing sessions.

— Mechanism described, no command or config key named desktop-v0.26.1

08 **Shell-command gating for scheduled tasks** NEW

45

Qwen Code gates scheduled tasks on a shell command run outside the model, and carries that gate through the daemon and Web Shell.

— Mechanism named but no exact command or flag desktop-v0.26.1

09 **Scoped external memory import jobs** NEW

45

Qwen Code adds scoped external import jobs for CLI and HTTP, enabling memory imports without restarting sessions.

— Surfaces named (CLI and HTTP) but no exact command v0.26.1

10 **Per-group session isolation restored for qqbot** BREAKING

45

Per-group session isolation under thread scope is restored for qqbot, which may break setups that relied on the previous cross-group session sharing behavior.

— Behavior change named but no migration steps given v0.26.1

11 **Public subagent delegation by default** NEW

35

Qwen Code enables public subagent delegation by default, allowing managed agents to spawn and coordinate subagents without additional configuration.

— Default behavior described, no config surface named desktop-v0.26.1

12 **Hook skipping for unhandled Claude Code events** IMPROVED

35

Qwen Code adds hook skipping for Claude Code events that the current runtime does not handle, avoiding spurious hook invocations.

— Mechanism described, no named config or flag desktop-v0.26.1

13 **web_fetch tool in Workspace Sessions** NEW

35

Qwen Code adds the `web_fetch` tool to Workspace Sessions in the Managed Agent path.

— Named tool but no usage example given v0.26.1

14 **Capability parity matrix for Managed Agent path** NEW

30

Qwen Code adds a capability parity matrix against the Managed Agent path in the CLI.

— Bare description, no concrete surface named v0.26.1

└──▷ **BREAKING ON UPGRADE**

! Per-group session isolation under thread scope is restored for qqbot, which may break setups that relied on the previous cross-group session sharing behavior.

WAS THIS USEFUL? 👍 👎

Yuval Sarel cones

Feedback ⓘ v0.4.5 NOTES ↗ RANK

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

> cones v0.4.5 adds an inline session-details shortcut and Hebrew keyboard support for control shortcuts, while removing the old left-arrow expand behavior.

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

01 **Session details expand via ctrl+e** BREAKING 70

Pressing `ctrl+e` now unfolds a session's details inline under its row in the dashboard. This replaces the previous left-arrow behavior, which no longer expands details and is now reserved purely for navigation.

— Names exact key and behavior change, no example shown v0.4.5

THINNER COVERAGE BELOW

02 **Hebrew keyboard layout for shortcuts** NEW 45

Control shortcuts now work under Hebrew keyboard layout by reading the Latin key beneath each Hebrew key, so users typing in Hebrew can still trigger shortcuts.

— Explains mechanism but no concrete command example v0.4.5

└──▷ **BREAKING ON UPGRADE**

! The left-arrow key no longer expands a session's details in the dashboard; use `ctrl+e` instead.

WAS THIS USEFUL? 👍 👎

ogcode

Feedback ⓘ v0.51.0 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's headline addition this window is multi-agent fan-out via `teammate spawn/wait/stop` with per-child spend tracking, paired with a server-enforced file-change gate spanning `write`, `edit`, and `bash`. Alongside that: multi-range file reads via `ranges`, exclusive per-project server locks, automatic image downscaling, Terraform/HCL code-map support, new config keys and API endpoints, and workspace theming and sidebar UI updates.

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

01 Multi-range file reads via ranges parameter BREAKING

95

Adds a `ranges` parameter to the `read` command, accepting a list of `{start_line, end_line}` objects so multiple non-contiguous file regions are fetched in a single call, each returned under its own `file:start-end` heading with numbered lines; `file_map`, `memory_map`, and the codemap renderer now advertise this form wherever a read window is suggested. The legacy top-level `start_line` / `end_line` pair and `offset` / `limit` are now refused by name when `ranges` is present (rather than silently ignored), so any call relying on the old silent-drop behaviour will error.

Read multiple non-contiguous regions of a large file in one agent step instead of five sequential calls.

```
$ read({"path": "src/server.go", "ranges": [{"start_line": 10,
"end_line": 40}, {"start_line": 120, "end_line": 155},
{"start_line": 300, "end_line": 320}]})
```

— Runnable example plus exact breaking behaviour spelled out v0.50.0

02 Custom OpenAI-compatible providers in ogcode.json NEW

90

Adds named OpenAI-compatible providers to `ogcode.json` via entries with `type`, `baseUrl`, and optional `model` and `label` fields; a `reloadProviders` call picks up edits live and the model picker groups entries by label.

Register a self-hosted or third-party OpenAI-compatible model so it appears in the model picker without restarting the server.

```json
{
  "providers": [
    {
      "type": "openai-compatible",
      "baseUrl": "https://my-llm-proxy.internal/v1",
      "model": "mistral-large",
      "label": "Internal Mistral"
    }
  ]
}
```

— Config fields and a runnable example provided v0.51.0

03 **Multi-agent fan-out with teammate spawn/wait/stop** NEW

88

Adds `teammate spawn` to fan work out from a parent build-agent session into named child streams, each with its own context window, spend ledgers, and uncapped step count; `teammate wait` blocks until one or more children finish, and `teammate stop` cancels a child and everything it spawned.

— Full mechanism and named subcommands, no runnable example given v0.51.0

04 **Exclusive server lock per project directory** NEW

88

Adds an exclusive server lock per project directory: running `ogcode serve` a second time against the same directory now exits immediately with a message naming the PID and address already serving it, with the lock held at `.ogcode/server.lock`.

Detect and prevent a duplicate server on a project directory — the second invocation exits with a clear message instead of silently corrupting state.

```
$ ogcode serve ./my-project
# In a second terminal:
ogcode serve ./my-project
# => exits immediately: 'directory already served by PID <n> at
<addr>' (lock held at ./my-project/.ogcode/server.lock)
```

— Exact command, path, and failure message shown v0.49.0

05 **File-change gate for write, edit, and bash tools** NEW

83

Adds a `GET /api/file-changes` endpoint serving per-file edit history (ring-capped, stored in the project database) to back a file-change gate that refuses a `write` or

`edit` when another session modified the file since the last read, returning the diff as a tool result; the same protection is applied to the `bash` tool, which fingerprints known files after every command and blocks the session's own shell on an unattributed change until the file is re-read.

— Endpoint and tools named, but no usage example v0.51.0

06 **previewDomain config key in ogcode.json** NEW 83

Adds `previewDomain` key to `ogcode.json` as the file-config equivalent of `OGCODE_PREVIEW_DOMAIN` for setting the live-preview domain; the environment variable takes precedence when both are set.

Pin a custom preview domain in your project config so every developer on the repo gets the same live-preview URL without setting an environment variable.

json

```
{
  "previewDomain": "preview.mycompany.internal"
}
```

— Exact config key and runnable example provided v0.47.0

07 **OGCODE_AUTO_INDEX_MAX_FILES indexing limit** NEW 83

Introduces `OGCODE_AUTO_INDEX_MAX_FILES` (default 500) to gate automatic background indexing on large projects — projects exceeding the limit prompt once per project before indexing.

Raise the auto-index ceiling on a large monorepo so Ogcode indexes up to 2000 files per turn without prompting.

```
$ OGCODE_AUTO_INDEX_MAX_FILES=2000 ogcode
```

— Exact env var, default value, and runnable example v0.47.0

08 **Workspace theming, session indicators, resizable sidebars, pinned sessions**

NEW 75

Adds a Theme control in Settings → General with Light, Dark, and System options — System follows the OS live — with the choice stored server-side so it applies across every browser tab and window. Adds live session-state indicators in the sidebar (a pulsing dot while a task runs, red on failure, absent when idle) tracked server-side to survive reloads. Makes both sidebars resizable by dragging the edge or nudging with arrow keys, remembering the chosen width across sessions, and adds the ability to pin sessions into a dedicated group at the top of the sidebar list, above the by-date groups.

Set the app to follow your OS light/dark mode so it shifts automatically when your machine switches themes in the evening.

> 📍 In the app, go to Settings → General → Theme and select `System`.

— UI paths named for each addition, one example given `v0.48.0`

---

09 **GET /api/config OS-reporting endpoint** NEW 75

Adds `GET /api/config` endpoint that reports the host operating system, enabling the web UI to surface only the search backends the current machine supports.

Query the host OS at runtime to decide which search backends are available before configuring automated search workflows.

```
$ curl -s http://localhost:<port>/api/config | jq '.os'
```

— Exact endpoint plus runnable curl example `v0.47.0`

---

10 **Terraform and HCL support in code map** NEW 73

Adds Terraform and HCL file support (`.tf` , `.tfvars` , `.hcl`) to the code map with a real outline, naming each block as it is addressed (e.g. `resource "aws_instance" "web"` becomes `aws_instance.web`).

— Exact file types and naming scheme shown, no command `v0.47.0`

---

11 **Automatic image downscaling on attachment** NEW 68

Attached images are automatically downscaled at attachment time: long edge capped at 1024 px, re-encoded as JPEG (for JPEG sources) or WebP otherwise, keeping screenshot text legible and reducing billed image tokens.

— Exact thresholds and formats, no user-facing command `v0.49.0`

---

12 **Windows support for deep-search browser fallback** IMPROVED 68

Extends the deep-search browser fallback to Windows, driving a private headless Chrome or Edge against a throwaway profile (previously macOS/Safari only); Settings → Search updates the engine-leads choice per platform.

— Mechanism and UI path named, no runnable example `v0.47.0`

---

13 **Teammate prompt mirroring to parent composer queue** NEW 65

Mirrors a teammate's `ask_user` questions and permission requests to the parent session's composer queue, tagged with the originating child, so the parent can answer

without switching sessions; a reply clears every copy of the prompt across both queues.

— Clear mechanism described, no UI path or example v0.51.0

14 **Teammate spend visibility in usage endpoints** NEW 62

Extends the parent session's token and usage endpoints with a `teammates` array so input, output, cache, and utility spend from all child sessions is visible alongside session-only totals.

— Named response field but no endpoint path or example v0.51.0

15 **Description field now required on write/edit calls** BREAKING 60

Every `write` and `edit` tool call now requires a `description` field (used as the history note); calls without it will be rejected.

— Exact field named, no migration guidance given v0.51.0

16 **Oversized attachment request rejection** NEW 60

Oversized requests are now measured and refused before reaching the OGX gateway, with a message naming the size, the 4 MiB limit, and the attachments responsible.

— Limit named, no example command shown v0.49.0

THINNER COVERAGE BELOW

17 **Per-turn resource telemetry** NEW 57

Reports a resource event at the end of every turn covering wall time, resident memory, peak and average CPU, goroutine counts, and host core and memory specs.

— Metrics named but no access method shown v0.50.0

18 **Agent-spawned processes isolated in their own session** IMPROVED 55

Processes started by the agent — shell commands and MCP servers — now run in their own session, preventing spawned processes from hijacking the terminal Ogcode was launched from.

— Mechanism named, no config or command shown v0.47.0

19 **Bundled Go skills from cc-skills-golang** NEW 50

46 Go skills from `samber/cc-skills-golang` are installed under `.agents/skills`, pinned in `skills-lock.json`.

— Named source and paths but no usage shown v0.49.0

└──▷ BREAKING ON UPGRADE

! Every `write` and `edit` tool call now requires a `description` field (used as the history note); calls without it will be rejected.

! The top-level `start_line` / `end_line` parameters on `read` are now refused by name (rather than silently ignored) when `ranges` is also supplied — any call that relied on the old silent-drop behaviour will now return an error.

! Combining `ranges` with `offset` / `limit` on `read` is now refused — any call that previously passed both is now an error.

WAS THIS USEFUL? 👍 👎

zot

Feedback ⓘ v0.4.48 NOTES ↗ RANK

Yet another coding agent harness, lightweight and written in go.

> zot's headline addition this window is a durable continuous-mode agent runtime — `zot continuous run` / `serve` / `attach` with journal, SQLite or PostgreSQL storage, hot backup and a crash-recovering task scheduler — alongside a string of TUI usability polish items and smaller agent-execution features like codemode orchestration and JSON model export.

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

01 **Continuous mode durable agent runtime** NEW 93

Adds `zot continuous run`, `zot continuous serve`, and `zot continuous attach` subcommands for persistent, crash-recoverable, collaborative agent sessions backed by a local journal, SQLite, or PostgreSQL (`--backend postgres`). Adds a stdio worker command with repository-scoped tools under the `continuous` subsystem, hot backup of open journal and SQLite stores so in-flight state is preserved without stopping the agent, extension support, committed observation, and queue placement, and routes all continuous work through a durable task scheduler so interrupted tools are reported rather than silently replayed. `zot continuous attach` connects the TUI to an already-running conversation, enabling live interaction with in-progress sessions.

Start a persistent agent server that survives crashes and accepts multiple clients — useful as a long-running background agent in a project repo.

```
$ zot continuous serve --backend postgres
```

Attach to an already-running continuous session from a second terminal to observe or steer it live.

```
$ zot continuous attach
```

Run a one-shot durable agent session locally, with crash recovery and task persistence written to the default journal store.

```
$ zot continuous run
```

Reconnect your terminal UI to a long-running continuous session that was started in another shell or detached context.



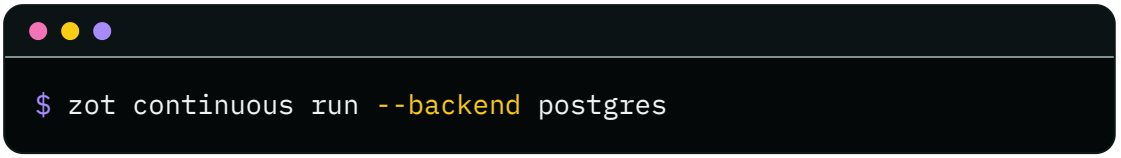
— Full subcommand set, backends and mechanism, with runnable examples v0.4.21 v0.4.45

02 **PostgreSQL backend and title persistence for continuous sessions**

IMPROVED 75

Adds `--backend postgres` PostgreSQL backend for `zot continuous` sessions, enabling externally hosted durable conversation stores, and persists conversation titles across storage backends in `zot continuous`.

Run a durable continuous session backed by an external PostgreSQL server instead of a local journal or SQLite file.



— Named flag and runnable command, but a thin feature on its own v0.4.35

THINNER COVERAGE BELOW

03 **model.list added to continuous host protocol** NEW 55

Adds `model.list` to the host protocol in continuous mode, enabling clients to query available models over the continuous run interface.

— Named protocol method but no usage example or client detail v0.4.25

04 **JSON export of picker models** NEW INTEGRATION NEW 50

Adds JSON export of picker models after discovery, enabling programmatic consumption of available model listings.

— Describes output format but no command or field names v0.4.41

05 **Codemode tool orchestration and reusable VM** NEW 45

Adds codemode tool orchestration and a reusable VM for coordinating agent tool execution.

— States the mechanism (VM, orchestration) but no commands or scope v0.4.15

06 **TUI display and interaction improvements** IMPROVED 40

A series of TUI polish items: keeps startup information displayed below the welcome banner on launch; renders codemode scripts alongside results in the TUI; adds thumbnail previews for pasted images; adds configurable navigation keybindings; adds

optional click placement and persistent text selection; and adds opt-in Markdown rendering for tool results.

— Each item named but only one-line descriptions, no mechanism or config v0.4.48

v0.4.38 v0.4.33 v0.4.30 v0.4.22 v0.4.17

07 **Explicit file content inclusion and Swarm image prompts** NEW

25

Adds explicit file content inclusion and Swarm image prompts to the coding agent harness.

— Bare name, no mechanism or usage detail given v0.4.26

NEW SUPPORTED PLATFORM MiniMax M3.1 Flash model support

NEW INTEGRATION deepseek-v4.1-flash added to opencode-go catalog

UPDATED INTEGRATION opencode-go model catalog synced with live endpoints

NEW INTEGRATION Claude Sonnet 5.5 and Haiku 5.5 in Bedrock catalog

UPDATED INTEGRATION Amazon Bedrock model catalog — Claude Opus 5.5

v0.4.37 NOTES ↗ STABLE

zot v0.4.37 sets MiniMax M3.1 Flash as the preferred model for the MiniMax provider while retaining legacy MiniMax models.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.4.37
https://github.com/patriceckhart/zot.git
# already have the repo? check out this version:
$ git checkout v0.4.37
```

v0.4.34 NOTES ↗ STABLE

Sends `x-opencode-session` header on opencode-go requests; adds optional cohesivity template to mcp-bridge.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.4.34
https://github.com/patriceckhart/zot.git
# already have the repo? check out this version:
$ git checkout v0.4.34
```

Sends `x-opencode-session` header on `opencode-go` provider requests for session-aware routing.

Adds optional cohesivity template to `mcp-bridge`.

v0.4.20 NOTES ↗ STABLE

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.4.20
https://github.com/patriceckhart/zot.git
# already have the repo? check out this version:
$ git checkout v0.4.20
```

v0.3.98 NOTES ↗ STABLE

zot v0.3.98 adds live model discovery for Amazon Bedrock, surfacing available models dynamically instead of relying on a static catalog.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.3.98
https://github.com/patriceckhart/zot.git
# already have the repo? check out this version:
$ git checkout v0.3.98
```

WAS THIS USEFUL? 👍 👎

ai-rulez

Feedback ⓘ v5.6.3 NOTES ↗ RANK

Author, validate and govern AI agent rules, skills, and MCP configuration across 52 coding assistants. Review with model evidence, evaluate skills, and publish bundles.

> Across eight releases, ai-rulez consolidated MCP skill-serving into a daemon-backed unified server with org-level read-only governance, added marketplace and plugin import from Claude Code, Codex and Cursor, introduced structured lint finding acceptance and tracked-only suppression, and layered in a large set of plugin-authoring, header, baz-preset and scanner-performance configuration keys.

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

01 MCP skill serving, unified toolset, and daemon mode NEW

92

Adds `mcp --serve-skills --discover-skills[=repo|all]` and `[mcp] discover_skills` to serve skills already in the repository via `skill://` resources, `find_skill`, and `load_skill`, with `repo` indexing `skills/` and `.claude/skills` and `all` adding installed and included skills. A later release merges MCP config-authoring tools and skill-serving tools into one process: `ai-rulez mcp --serve-skills` registers `find_skill`, `load_skill`, `skill://` resources, and config tools under a single `ai-rulez` entry; `--read-only` registers only read-only tools; `AR993` counts the single `.mcp.json` entry. Adds `ai-rulez mcp --serve-skills --daemon` to share one long-lived per-repository server via a `0600` socket in a `0700` user-cache directory keyed by repo toplevel, ai-rulez version, and serving flags; the daemon exits after 15 minutes idle, reloads the catalog on change, and is not available on Windows.

Start a single long-lived MCP server that serves both skill-browsing tools and config-authoring tools for the repo — avoids running two separate processes.

```
$ ai-rulez mcp --serve-skills --daemon
```

— Names sockets, cache dir, timeouts, codes across both releases. v5.0.4 v5.5.0

02 External scanner batching, caching, and scoping IMPROVED

92

Adds `batch_size` and `parallel` (1–16, default 1) on `[[lint.external]]` to split and concurrently run scanner batches, and `[lint.scanner_policy] max_timeout` to raise the per-run ceiling from 15 minutes up to 6 hours. Adds `--no-scan-cache` and per-skill scanner result caching keyed by skill tree hash, plus `--changed` and `--since` flags to narrow scans to only skills with changed files. Adds `stage_globs` and `stage_exclude` on `[[lint.external]]` to control which files of a skill directory

reach external scanners (previously only `SKILL.md`, `scripts/`, `references/`, and `assets/` were staged), and a `cache_files` list plus `version_command` field to key the scanner cache on script content and check version ranges against the scanner binary rather than its interpreter.

Run only changed skills through external scanners in batches of 10, with up to 4 concurrent batches, to stay well within CI time budgets.

```
$ ai-rulez validate --external --changed
```

— Names every flag/key and default/limit values with an example. v5.5.0

---

03 **MCP server configuration and read-only governance** NEW

90

Adds an `[mcp]` config block in `ai-rulez.toml` (and layered/local configs) with `read_only`, `daemon`, `no_watch`, `allow`, `deny`, and `tools = [...]` keys mirroring their CLI flag equivalents, so `.mcp.json` can invoke `ai-rulez mcp` without extra flags; a CLI flag wins over config. Adds a `--tools` flag to `mcp` and `mcp --serve-skills` to register only a named subset of tools, with an unknown name stopping the server and listing valid ones. Organization policy can enforce `[mcp] read_only = true` so no repository key or CLI flag can loosen it; setting `read_only = false` in a repository is reported as `AR740`. Adds an `mcp --exclude <glob>` flag and `[mcp] exclude` config key, which appends to the configured list, to keep paths out of repository discovery and the file watcher.

Lock down an org MCP server to read-only at the policy level so no repository or flag can widen access.

```toml
[mcp]
read_only = true
```

Keep generated output directories and vendor paths out of MCP skill discovery and the file watcher.

```
$ ai-rulez mcp --exclude 'dist/**' --exclude 'vendor/**' --
serve-skills
```

— Names every key, flag, and the AR740 enforcement code. v5.5.1

---

04 **Pi plugin naming, skills layout, and drift verification**

IMPROVED

90

Adds `package_name` to `[plugin.pi]` to set the npm package name instead of the default `@<owner>/pi-<plugin-name>` pattern, and `skills_entry = "directory"` to write `"./skills"` to `pi.skills` and `"skills/"` to `files` instead of one entry per skill. Adds `[plugin] skills_dir_owned = true` to declare the bundle's skills directory fully owned, so `verify --plugin` reports ungenerated files and directories as drift of kind `extra` and `generate --plugin` removes them. Enhances `verify --plugin` to name each drifted output with its kind (`changed` , `missing` , `stale-source` , `obsolete` , `extra` , `symlink`), carrying a `drift` array of `{path, kind}` objects under `--format json` .

Use a custom npm package name for a Pi plugin instead of the auto-generated `@<owner>/pi-<plugin-name>` default.

```toml
[plugin.pi]
package_name = "@myorg/my-custom-plugin"
```

Declare the skills directory fully owned so stale generated files are surfaced as drift and removed on next generate.

```toml
[plugin]
skills_dir_owned = true
```

Audit a plugin bundle for drift and get a machine-readable list of every drifted path with its kind.

```
$ ai-rulez verify --plugin --format json
```

— Names all config keys, drift kinds, and runnable examples. v5.1.0

05 **Structured lint finding acceptance** NEW 86

Adds `[[lint.accept]]` entries in `.ai-rulez/config.toml` to accept individual findings by `rule` , `path` glob, `match` substring, and/or `line` , each requiring a mandatory `reason` ; accepted findings stay in the report but no longer fail the run. A new AR9I0 accept-unused warning fires when an acceptance entry matches nothing, surfacing stale acceptances instead of letting them accumulate silently.

Suppress a known finding in CI without removing it from the report — accept exactly one instance with a documented reason so the acceptance itself is auditable.

```toml
# .ai-rulez/config.toml
[[lint.accept]]
rule = "AR051"
path = "src/cache/**"
match = "__pycache__"
reason = "Build artifact produced by the packaging step; not source code."
```

— Names fields, warning code, and gives a config example. v5.6.2

06 **Plugin authoring: shared resources, provenance, and versioning**

NEW

86

Adds `[plugin] shared_resources` to copy shared directories (e.g. `references/`) into every bundled skill for every runtime, rewriting relative links from `SKILL.md` and recording copies in the provenance sidecar. Adds `[plugin] provenance_headers = false` to write bundled Markdown/JS/TS/PY files byte-identical to their sources while still recording outputs in `.ai-rulez-generated.json`. Adds `[plugin] version_from` (accepting `"package.json"`, `"pyproject.toml"`, or `"Cargo.toml"`) to read the plugin version from an existing manifest's `project.version` or `package.version`, mutually exclusive with `version`. Adds `[plugin.pi] skills_dir` to write each Pi-runtime skill once into a shared tree and point `pi.skills` at it, plus `package_json = "merge"` to update only ai-rulez-owned keys in `package.json`, preserving `scripts`, `devDependencies`, and other fields. Adds `[plugin] package_manager` (emits `packageManager` in generated OpenCode/Pi root `package.json`) and `[plugin] opencode_dependency` (overrides the `@opencode/plugin` constraint, default bumped to `^2.0.26`).

— Names every plugin config key and default, no runnable example. v5.0.4

07 **Recursive generate/validate directory exclusions** NEW

84

Adds a `recursive_exclude` config key (e.g. `recursive_exclude = ["tests/fixtures"]`) to `.ai-rulez/config.toml`, `.agents/ai-rulez.toml`, or `.config/ai-rulez/config.toml`, plus a repeatable `--exclude` flag on `generate` and `validate`, to skip named directories when running with `--recursive`. The `generate_outputs` MCP tool now honours both the `recursive` and `exclude` parameters, matching CLI behaviour.

Exclude test-fixture directories from a recursive generate run so repos with deliberately invalid configs don't fail CI.

```
$ ai-rulez generate --recursive --exclude tests/fixtures --
exclude testdata
```

Persist fixture exclusions in config so every recursive run (CI and local) skips them automatically.

```toml
recursive_exclude = ["tests/fixtures", "testdata"]
```

— Exact config paths, flag, and runnable examples given. v5.6.3

08 **Tracked-only lint suppression** NEW 80

Adds `--tracked-only` flag to `validate` and `scan`, and the equivalent `[lint] tracked_only = true` config key, to suppress findings in files outside the git index (e.g. untracked build artifacts) while still walking and processing them.

Run validation in CI so that untracked build artifacts on a fresh checkout never produce spurious findings that differ from a developer's machine.

```
$ ai-rulez validate --tracked-only
```

Permanently opt a project into tracked-only linting so every `validate` and `scan` invocation omits untracked files without requiring a flag.

```toml
# .ai-rulez/config.toml
[lint]
tracked_only = true
```

— Exact flag, config key, and two runnable examples. v5.6.2

09 **Lint diagnostics for skill pointers and root shipping** NEW 78

Adds `AR990`, which flags path-style and bare-name pointers to served skills (e.g. `.ai-rulez/domains/<d>/skills/<name>/SKILL.md`, hyphenated bare names like `see skill dev-cycle`) and suggests `load_skill <name>` as the fix. Adds `AR9T0`, a root-component-shipped warning raised when a single-plugin root ships paths that hosts load by default (`.mcp.json`, `hooks/`, `agents/`, `commands/`, `settings.json`, …), naming the host and suggesting `content_root` or a monorepo member as the fix.

— Names both rule codes and their suggested fixes, no example. v5.0.4

10 **skip_outputs to protect hand-written files** NEW 76

Adds a `skip_outputs` config key (also accepted in `config.local.toml`) listing output kinds — `instructions`, `agents_md`, `claude_md`, `rules`, `skills`, `agents`, `commands` — that ai-rulez never writes, overwrites, or deletes.

Keep certain output files hand-written and prevent ai-rulez from overwriting them on generate.

```toml
skip_outputs = ["claude_md", "agents_md"]
```

— Lists every output kind and a config example. v5.5.0

11 **Header customization for nested root files** NEW 76

Adds `[header] nested` — `compact` (default), `omit`, or `inherit` — to shorten or suppress the header of a root file generated inside another root, while preserving the Content-Hash line so `generate --check` stays valid. Adds `[header] mcp_line = false` to drop the MCP server line from generated headers.

Trim the verbose banner when a nested project root is generated inside a monorepo root, keeping headers short without losing the Content-Hash line.

```toml
[header]
nested = "compact"
```

— Names config keys, option values, and a config example. v5.0.4

12 **Baz preset rule targeting controls** NEW 76

Extends `targets` to accept negations (e.g. `targets: ["!baz"]`) so a rule can be excluded from a single preset; adds `[rules.baz] include = [...]` and `baz: true` frontmatter for per-rule opt-in under `baz_default = "none"`. Adds `[rules] baz_unscoped = "skip"` to exclude directory-less path-scoped rules (e.g. `**/*.go`) from the root `AGENTS.md` when `baz` is the only preset writing that file (default remains `"root"`), and `[rules] baz_dedupe_max = K` to collapse a rule copied into more than K directories into a single copy at the nearest covering directory.

— Names every key and default value, no runnable example. v5.0.4

13 **Control over unlisted-skills instructions text** NEW 72

Adds `skills_instructions = false` and `skills_instructions_text` (supporting `{{server}}` and `{{skills_dir}}`) under `[mcp]` to suppress or customise the

`Unlisted skills` section injected into `AGENTS.md` and other MCP-capable harness root files while skills are served.

Suppress the auto-injected `Unlisted skills` section from AGENTS.md and all harness root files, or replace it with a custom message referencing the MCP server name.

```toml
[mcp]
skills_instructions = false
```

— Names config keys, template variables, and a config example. v5.0.4

14 **Marketplace and plugin import from coding assistants** NEW

70

Adds `[[installed_marketplaces]]` and `[[installed_plugins]]` config blocks to import plugins and marketplaces from Claude Code, Codex, and Cursor, alongside new commands `ai-rulez marketplace add|remove|list|update` and `ai-rulez plugin install|remove|list|update`.

— Names config blocks and commands but no runnable example given. v5.5.0

15 **Custom dynamic-skills stub content** NEW 70

Adds a `[delivery.stub]` config block to customise the generated `dynamic-skills` stub: `description`, `body`, and `body_file` (relative to config dir) replace stub text and may use `{{server}}`; `validate` enforces a non-blank description, exclusive body forms, known variables, and a reachable file.

— Names all keys and validation rules, lacks a worked example. v5.0.4

16 **Automatic generate on remote session start** NEW 66

Adds a `[session_start]` config block and `ai-rulez session-start` command to automatically run `ai-rulez generate` when a remote cloud session starts, configured via keys `generate`, `when`, `mode`, `timeout`, `version`, `command`, `prompt`, and `prompt_text`.

— Names command and all config keys but no example invocation. v5.5.0

17 **Reserved-name validation for skills and commands** NEW

64

Adds an `AR808 reserved-name` validation rule to `validate` that errors when a skill or command name (including `name:` frontmatter) collides case-insensitively with a Claude Code built-in command or bundled skill; intentional overrides are acknowledged via `lint.allow_overrides`.

— Names the rule code and override key, no example given. v5.5.1

18 **Refresh remote includes before generate** NEW 64

Adds a `generate --refresh-remote` flag (and `[lock] refresh_on_generate = true` config key) to re-resolve remote includes, installed skills, and skill sources to their newest allowed version before writing outputs.

— Names flag and config key; no example shown. v5.5.0

19 **GitHub Action LLM inputs and stable v5 tag** IMPROVED 62

The ai-rulez GitHub Action adds `llm-*` inputs covering all 16 supported LLM environment overrides, with settings persisting to later workflow steps. The floating `Goldziher/ai-rulez@v5` action tag now only advances after successful GitHub, npm, PyPI, and Homebrew publications, so prereleases or failed runs can't move it backward.

— Names the input pattern and tag but gives no example usage. v5.4.1

20 **Minimal project init without .ai-rulez folder** NEW 60

Adds `ai-rulez init --minimal` to create a root-level `ai-rulez.toml` / `ai-rulez.local.toml` configuration without a `.ai-rulez/` folder.

Bootstrap a lightweight project that manages only remote skills and plugins without a `.ai-rulez/` folder, then regenerate on every remote cloud session automatically.

```
$ ai-rulez init --minimal
```

— Exact flag and runnable command, minimal mechanism beyond that. v5.5.0

THINNER COVERAGE BELOW

21 **migrate v5 command for config upgrade** NEW 50

Adds a `migrate v5` command to convert a 4.x configuration to v5 in place.

— Names the exact command but no further mechanism given. v5.0.0

22 **Python dependencies for Hermes plugins** NEW 28

Adds Python dependency declaration for Hermes plugins via the plugin authoring system.

— Thin description with no config key or example named. v5.0.0

NEW DETECTION Additional credential-exfiltration and prompt-injection patterns

> v5.4.0 NOTES ↗ STABLE
>
> v5.4.0 overhauls the judge command with evidence-verified reviews, SARIF/JSON v2 output, and `--fail-on`; adds secret-screening on outgoing model messages; and

refreshes Bedrock credentials across clients.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v5.4.0 https://github.com/Goldziher/ai-rulez.git
# already have the repo? check out this version:
$ git checkout v5.4.0
```

Replaces the `judge` command's `--rubric` and `--min-score` flags with rubric skills and `--fail-on`; the command now uses verified findings with quoted evidence rather than v1 per-file score reports.

Changes `--changed` base syntax to `--changed=REV`; omitting `--changed` replaces `--changed=false`.

Adds JSON v2 and SARIF report output formats for judge reviews.

Adds security, governance, and quality reviewer roles to the judge command that verify quoted evidence and adjudicate findings.

Adds Bedrock credential refresh across clients without taking ownership of user-supplied static credentials.

+3 more

└──▷ BREAKING ON UPGRADE

! The `judge` command's `--rubric` and `--min-score` flags are removed; replace them with rubric skills and `--fail-on`.

! For `--changed`, specify an optional base as `--changed=REV`; omit the flag entirely instead of passing `--changed=false`.

WAS THIS USEFUL? 👍 👎

Diagram Design

Feedback

changes since 2026-10-10

CODE ↗

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

Adds service blueprint visual type and architecture containment scopes to Diagram Design.

.

└──▷ TRY IT

Verify a service blueprint diagram enforces correct layer ordering, visibility-line placement, and stage-grid alignment before export.

```
$ python3 scripts/verify-service-blueprint.py <path-to-blueprint.html>
```

Adds `service blueprint` visual type with a dedicated verifier (`verify-service-blueprint.py`) that enforces layer order, visibility-line placement, stage-grid alignment, and cell census across customer, frontstage, backstage, and support layers using `data-layer` , `data-stage` , `data-name` , and `data-role="visibility"` attributes.

WAS THIS USEFUL? 👍 👎

Anthropic Claude Code

Feedback ⓘ v2.1.310 NOTES ↗

CODE ↗ RANK

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

> Claude Code v2.1.310 adds a Monitor tool for enterprise deployments on Bedrock, Vertex, and Foundry, expands Enterprise custom roles with cloud session and Remote Control permissions, and polishes cloud environment setup and Linux sandbox diagnostics.

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

01 **Sandbox AppArmor diagnostics on Linux** IMPROVED 80

Improves `/sandbox` and the sandbox startup warning on Linux to name Ubuntu's AppArmor restriction on user namespaces, and how to lift it, when it keeps bubblewrap from starting the sandbox.

— Names the exact command, OS, and underlying restriction mechanism v2.1.310

THINNER COVERAGE BELOW

02 **Monitor tool and plugin monitors on cloud platforms** NEW 55

Adds the Monitor tool and plugin monitors on Bedrock, Vertex, and Foundry, and in environments where telemetry or nonessential traffic is disabled; long foreground sleeps are now blocked in those environments.

— Names three platforms but no command or config surface given v2.1.310

03 **Cloud Sessions and Remote Control permissions for Enterprise roles**

NEW 55

Adds Claude Code cloud sessions and Remote Control permissions to Enterprise custom roles, so Owners can control access to each by role.

— Names the two permissions and who controls them v2.1.310

04 **Cloud environment setup script UX improvements** IMPROVED 55

Adds a Choose file button to load a cloud environment's setup script from a file, and improves the session start checklist to show the setup script's output in a scrolling box.

— Names the UI elements but gives no exact navigation path v2.1.310

WAS THIS USEFUL? 👍 👎

OpenQodex

Feedback ⓘ openqodex@0.14.0 NOTES ↗ RANK

Open source code review that runs in your coding agent before you push. Scanners plus a separate AI reviewer, on your own login, no key.

> OpenQodex 0.14.0 introduces a staged, multi-session review pipeline with a dedicated verifier pass, opens up a full public graph API for building and querying code graphs, and broadens lost-behavior detection to dozens more languages, alongside a mentioned but undetailed universal lockfile scanning capability.

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

01 **Staged multi-session review pipeline** NEW 85

Introduces staged review execution: two finder sessions read each changed file with surrounding context, then a fresh verifier session confirms candidates, and only confirmed candidates become findings; style, naming, doc, and test items surface separately as observations. The `review.stages: false` config key reverts to the original single-pass review mode.

— Explains mechanism and gives exact config key to opt out. openqodex@0.14.0

02 **Public graph API for building and persisting code graphs** NEW 80

OpenQodex now exports `query`, `querySliced`, `OPERATIONS`, `SLICE_CHECKS`, `parseTarget`, `resolveTarget`, `prepareIndexes`, `writeIndex`, `readIndex`, `readIndexFrom`, `assemble`, `INDEX_FORMAT`, `serializeModel`, `deserializeModel`, `graphOf`, `floorReasons`, `hotSymbols`, and `isTestPath` from the library, letting external programs build a graph, persist it, reload it, and run every question over the result. `writeIndex` now also includes `projects.json` in its returned file set, making those files self-sufficient without additional project metadata.

— Names every exported function and the exact file-set change. openqodex@0.14.0

03 **Lost-behavior detection extended to dozens of languages**

IMPROVED 65

Extends lost-behavior detection for replaced lines to cover `.c`, `.cc`, `.cpp`, `.cxx`, `.C`, `.h`, `.hh`, `.hpp`, `.hxx`, `.rs`, `.cs`, `.php`, `.ts`, `.tsx`, `.js`, `.jsx`, `.mjs`, `.cjs`, `.mts`, `.cts`, `.py`, `.pyi`, `.go`, `.rb`, `.java`, `.kt`, `.kts`, `.swift`, `.m`, `.mm`, `.scala`, `.sc`, `.sh`, `.bash`, `.zsh`, `.sql`, `.lua`, `.pl`, `.pm`, `.r`, `.R`, `.dart`, `.ex`, `.exs`, `.erl`, `.hrl`, `.hs`, `.clj`, `.cljs`, `.cljc`, `.vue`, `.svelte`, `.vb`, `.fs`, `.fsx`, `.ps1`, `.psm1`, `.groovy`, `.gradle`, `.asm`, `.s`, `.S`, `.zig`, `.jl`, `.ml`, `.mli`, `.nim`, `.v`, `.vhd`, `.vhdl`, `.f`, `.f90`, `.f95`, `.for`, `.cob`, `.cbl`, `.coffee`, `.elm`, `.el`, `.lisp`, `.lsp`, `.scm`, `.ss`, `.rkt`, `.d`, `.pas`, `.p`, `.raku`, `.tcl`, and `.awk`.

— Full extension list given but no mechanism beyond coverage expansion. openqodex@0.14.0

THINNER COVERAGE BELOW

04 **Universal lockfile scanning** NEW 10

OpenQodex 0.14.0 adds universal lockfile scanning, per the release summary; no further detail on supported formats, lockfile types, or mechanism is provided.

— Mentioned only in summary, no mechanism or scope given. openqodex@0.14.0

WAS THIS USEFUL? 👍 👎

Tester Army e2e

Feedback ⓘ @e2e-dev/testmu@0.0.2 NOTES ↗

RANK

Next generation e2e testing framework for web and mobile apps.

> This window's headline is AI-driven change-based test filtering via `decisionFilter` and `--changed-since`, alongside three new test-target providers (TestMu AI, Lightpanda, Limrun), a programmatic `e2e/runner` host API, `e2e run --watch`, and a long run of CLI, mobile, and web-engine additions across 18 releases.

└──▷ **WHAT SHIPPED · 43 FEATURES** most completely described first ? what's the number?

01 **AI-driven change-based test filtering** NEW 96

`@e2e-dev/github` adds a `--changed-since <ref>` flag (e.g. `--changed-since origin/main`) to scope CI runs to tests affected by a change, and an `e2e list --changed-since origin/main` subcommand to preview which tests would be skipped. A `filters: [decisionFilter({ model })]` test-config entry (from `@e2e-dev/decision`) asks an AI decision model which tests a pull request can affect, e.g. `filters: [decisionFilter({ model: gateway.decisionModel('typesafe-ai/jev') })]`; the model must answer `boolean` questions and `providerOptions` passes call-level options such as zero data retention to every invocation. Skipped tests are surfaced in the pull request comment, `junit.xml`, and the report; a filtered run with no tests to execute starts no engine or app and the CLI exits as soon as output is written after Ctrl-C or a reporter deadline; and when the loaded config file itself has changed (readable as `context.configFile`), `decisionFilter` skips no tests, e.g. after editing `e2e.ci.config.ts`. Requires `e2e` 0.21.3 or later.

— Full mechanism, exact flags, and config keys all named. @e2e-dev/github@0.4.3 @e2e-dev/decision@0.3.0

02 **Programmatic runner API (e2e/runner)** NEW 88

run() from `e2e/runner` executes suites with a config file or runtime configuration, streams progress, can cancel execution, and returns the completed report without exiting the caller; calling `openHost` while a suite is running fails early with `RUNNER_BUSY`. openHost<Fixtures>() provides a worker-lifetime host that starts app processes and initializes the engine once; host.openSession({ name, timeout, signal, agent, onStep }) opens a per-test attempt with typed fixtures, its own deadline, steps, and artifacts, writing a redacted `trace.md` under `results/host/` with an absolute `tracePath` on close.

— Full function signatures and error codes documented. e2e@0.21.1 e2e@0.21.0

03 **e2e init scaffolding and non-interactive flags** NEW

85

`e2e init --example <name>` scaffolds a new project from a named example app (`with-vite`, `with-next`, `with-astro`, `with-expo`, `with-swiftui`, `with-compose`, `with-kotlin-multiplatform`, `with-flutter`, `with-maui`). `--engine`, `--gateway`, `--endpoint`, `--skills`, `--mcp`, `--install` / `--no-install`, and `--yes` flags skip individual wizard prompts non-interactively; `--endpoint` implies `--gateway openai-compatible`, `--skills` and `--mcp` accept comma-separated locations or `none`, and an unknown value exits 2 before any write.

— Every flag, template, and exit behavior named. `e2e@0.21.3` `e2e@0.21.0`

---

04 **e2e run --watch interactive mode** NEW

83

`e2e run --watch` (`-w`) re-runs tests automatically on changes to test files, source files, or config, with interactive keys: `a` `/enter` for all tests, `r` for current filters, `f` for failed, `u` to update snapshots, `p` / `t` to filter by file/title, `q` to quit; disabled in CI and with `--reporter json`.

— Flag, alias, and full key bindings all documented. `e2e@0.20.0`

---

05 **device.locator() role term alignment with agent-device** BREAKING

82

device.locator() and the `selector` locator now match using agent-device's own term rules (`listSelectorChainMatches`); `role` takes agent-device kinds (`text-field`, `navigation-bar`, `scroll-area`, `search`, `cell`) and text terms (`id`, `label`, `value`, `text`) compare case-insensitively. This is a breaking change: `role=textbox` must become `role=text-field`, `role=navigation` must become `role=navigation-bar`, `role=search-field` must become `role=search`, and `role=listitem` must become `role=cell` — selectors using the old role names will match different or no nodes.

— Exact migration mapping given for every renamed role. `@e2e-dev/mobile@0.12.0`

---

06 **Modifier keys for tap/double_tap/right_click tools** NEW

80

An optional `modifiers` list (Shift, Control, Alt, Meta, `ControlOrMeta`) is added to the `tap`, `double_tap`, and `right_click` tools in `agent.act` and the MCP server, matching the test API's tap({ modifiers }); modifiers are recorded and replayed from cache, and custom executors access the capability via `ctx.target.tapModifiers`, passing `{ modifiers }` to `ctx.actions.tap`, `doubleTap`, and `secondaryTap`.

— Every tool, modifier value, and extension hook named. `e2e@0.21.2`

---

07 **app.command.inheritEnv config key** NEW

78

`app.command.inheritEnv` config key passes host environment variables to the app by name or trailing-`*` prefix (e.g. `['DATABASE_URL', 'STRIPE_*']`), with `command.env`

taking precedence and inherited values redacted from startup failure output; a bare `*` or mid-string `*` yields `INVALID_CONFIG`.

— Exact config key, example values, and error code given. `e2e@0.21.0`

08 **`--max-duration` run time limit** NEW 78

`e2e run --max-duration <ms>` flag and `maxDuration` config key cap total run wall-time: running tests are marked `interrupted`, remaining tests are skipped with cause `time-limit`, and the run exits 1 with `RUN_TIMEOUT`; `--last-failed` retries interrupted and skipped tests.

— Exact flag, config key, exit code, and retry path. `e2e@0.21.0`

09 **`e2e feedback` reporting flags** NEW 73

`e2e feedback` adds `--error-code`, `--repro`, and `--allow-outdated` flags; `--repro` is enforced for `--type bug` reports (otherwise a prefilled GitHub issue URL is printed), the command exits 2 when the installed version is outdated, and `--task`, `--expected`, `--actual`, and `--approach` are capped at 1500 characters.

— All flags and validation limits named exactly. `e2e@0.21.0`

10 **`--action-timeout`/`--assertion-timeout` flags** NEW 72

`--action-timeout <ms>` and `--assertion-timeout <ms>` flags on `e2e run` override the config's `actionTimeout` and `assertionTimeout` for a single run (or every rerun in watch mode); values must be integers 1–2147483647 or exit 2 is returned.

— Exact flags, value range, and exit code named. `e2e@0.21.3`

11 **TestMu AI browser and device provider** NEW 70

`@e2e-dev/testmu` adds testmuBrowsers() for running web targets in TestMu AI hosted Chrome or Edge, one session per worker slot, and supports mobile testing on TestMu AI Android emulators, iOS simulators, and real devices. Attempts that record video automatically link TestMu AI's session recording, and `@e2e-dev/mobile` and `@e2e-dev/web` are now optional peer dependencies so a project installs only the engine it uses.

— Names the provider function and dependency change; no code sample.

`@e2e-dev/testmu@0.0.2` `@e2e-dev/testmu@0.0.1`

12 **iOS network and airplane mode capability change** BREAKING 70

device.setNetwork('offline') and device.setAirplaneMode(true) now fail with `UNSUPPORTED_CAPABILITY` on iOS; use `platforms: ['android']` with setAirplaneMode(true) (Android 11+) for offline tests. Note that setNetwork('offline') disables Wi-Fi only and does not cut cellular data.

– Exact migration path and caveat given. `@e2e-dev/mobile@0.12.0`

13 **Locator methods wait until action timeout** IMPROVED 68

textContent(), inputValue(), getAttribute(), isEnabled(), isDisabled(), isChecked(), and boundingBox() now wait for their matched node until the action timeout instead of failing immediately with `LOCATOR_NOT_FOUND` when the node is still rendering.

– Every affected method and prior error code named. `e2e@0.21.0`

14 **Limrun device provider** NEW 67

The Limrun device provider starts one hosted instance per worker, handles app installation, records sessions through agent-device, and cleans up on run exit. Retained driver leases renew automatically while other targets run, provider-owned local daemons stay isolated from ambient remote-daemon environment settings, and `agent-device` ships as a regular dependency pinned to the mobile engine's version, removing the need for manual installation.

– Several concrete mechanisms named, no config key or flag. `@e2e-dev/limrun@0.2.0`

15 **Lightpanda browser provider** NEW 65

lightpanda() is a new browser provider that starts a `lightpanda serve` process per worker slot on a free port and stops it automatically when the run ends.

– Named function and lifecycle described, no usage example. `@e2e-dev/lightpanda@0.0.1`

16 **Keyboard visibility detection enabling dismiss_keyboard** NEW 62

Executor observations now carry `keyboardVisible`, the engine's measured on-screen keyboard state, enabling the decision executor to offer `dismiss_keyboard` on mobile. On iOS, `XCUIElementType*` element types are mapped onto roles for agent-device WebDriver sessions, enabling `getByRole` queries and keyboard-visible detection.

– Names the field and action, mechanism spans two packages. `@e2e-dev/web@0.15.1` `@e2e-dev/mobile@0.12.2`

17 **e2e upgrade/update command** NEW 62

`e2e upgrade` (alias `e2e update`) upgrades `e2e` to the latest release, a dist-tag such as `nightly`, or a specific version, with `--dry-run` to preview the plan.

– Command, alias, and dry-run flag all named. `e2e@0.21.3`

18 **test_lines MCP catalog tool** NEW 62

A read-only `test_lines` MCP catalog tool returns the session's recorded test lines in order as a draft test body, and every `e2e mcp` action now replies with the corresponding test line. A project tool can no longer be named `test_lines` — the name is now reserved by this built-in tool.

– Tool name and breaking naming conflict both stated. `e2e@0.21.3`

19 **`locator.or()` union locator matching** NEW 60

locator.or(other) matches either locator in document order and stays strict when both match, available in both `@e2e-dev/web` and `@e2e-dev/mobile`. Third-party engines must handle the new `union` kind of `LocatorExpression` this introduces.

– Names the API and breaking LocatorExpression kind, no usage sample. `@e2e-dev/web@0.15.3` `@e2e-dev/mobile@0.12.4`

20 **Mobile device session persistence across worker replacement**

IMPROVED 60

Healthy agent-device sessions on leased devices stay open across worker replacement, preventing `'Device is already in use'` failures in later tests. Worker device selection is also forwarded across video recording, app state resets, device settings, alerts, and clipboard requests — including after device.closeApp() — so the chosen device persists when multiple simulators or emulators are booted.

– Clear before/after failure mode, no user-facing API named. `@e2e-dev/mobile@0.12.4` `@e2e-dev/mobile@0.12.2`

THINNER COVERAGE BELOW

21 **`device.fold` support for foldable Android emulators** IMPROVED 57

`device.fold` now works on foldable Android emulators, setting emulator console posture and clearing the lock screen raised on fold to the cover display; it previously failed with `UNSUPPORTED_CAPABILITY` on Android.

– Clear before/after and named API, no code sample. `@e2e-dev/mobile@0.12.0`

22 **`device.setPermission()` via agent-device before first launch**

IMPROVED 57

device.setPermission() and launch permission presets now set permissions via agent-device's `settings permission --app` before the app's first launch, without opening the app first, and always act on the pinned app.

– Named method and underlying command, no example output. `@e2e-dev/mobile@0.12.0`

23 **`screen.tree()` and `screen.debug()`** NEW 57

screen.tree() returns the full screen as the agent reads it (one node per line, secrets redacted); screen.debug() prints the same text to test output.

– Two named methods with clear output format. `e2e@0.21.3`

24 **`onAction` hook for `agent.act`** NEW 55

`onAction` callback on `agent.act` fires after each action once the screen settles — including replayed actions — enabling per-action screenshots.

— Named hook and trigger condition, no code sample. `e2e@0.21.3`

25 **OPENCODE_API_KEY support for opencode-console** NEW 53

`e2e models opencode-console` supports `OPENCODE_API_KEY`, listing models via the key in place of stored login credentials; `e2e models` with no provider includes OpenCode Console when only the key is set.

— Named env var and command, no further mechanism. `e2e@0.21.0`

26 **Editor plugin installation for Cursor, Claude Code, Codex, Grok Build**

NEW 53

The e2e plugin installs in Cursor from `~/.cursor/plugins/local/` with the same three skills as the Claude Code and Codex plugin; Grok Build reads the Claude Code files via the repository.

— Exact install path named, skills not enumerated. `e2e@0.21.3`

27 **Canvas field input via type_at** NEW 52

`type_at` fills a field drawn on a canvas: the text model names the field and its value, the screenshot locates it, and the keyboard types into it.

— Named command with mechanism, no runnable example. `@e2e-dev/decision@0.2.3`

28 **browser.keyboard.down/up for held keys** NEW 51

`browser.keyboard.down` and `browser.keyboard.up` methods support held keys, repeat events, and modifier sequences.

— Named methods with clear scope, no code sample. `@e2e-dev/web@0.15.1`

29 **Engine disposal revocable permission** NEW 50

Engine disposal can receive optional revocable permission to preserve run-owned resources for a replacement worker; final shutdown, forced termination, and parent disconnect revoke that permission while disposal is pending.

— Mechanism described but no named flag or key. `@e2e-dev/web@0.15.3`

30 **Decision executor tap accuracy and vision support** IMPROVED 50

`vision: true` is now compatible with TypeSafe Jev via `@ai-sdk/typesafe-ai` 3.0.19 or later, and `tap_at` accuracy is improved with a third, tighter look before tapping, so it reliably hits small drawn controls such as map pins.

— Names package versions and improvement, no config shown. `@e2e-dev/decision@0.2.3`

31 **`--shard` skipped-test dedup** IMPROVED 50

`--shard` now reports a skipped test in only one shard (the one running the nearest preceding test, or the first shard), eliminating duplicate skipped-test counts in summaries, `junit.xml`, and `report.json`.

— Exact rule and affected output files named. `e2e@0.21.2`

32 **Config file discovery for .js and .mjs** IMPROVED 48

Finds `e2e.config.js` and `e2e.config.mjs` in addition to `e2e.config.ts` and `e2e.config.mts`, so plain-JavaScript projects no longer need a `--config` flag.

— Exact filenames named, minor convenience change. `e2e@0.21.3`

33 **`browser.unrouteAll` route clearing** NEW 45

`browser.unrouteAll` clears every route registered through the browser fixture while preserving configured site headers.

— Named method with narrow scope, no example. `@e2e-dev/web@0.15.3`

34 **Longest-file-first worker scheduling** IMPROVED 45

`e2e run` with more than one worker now starts the files that took longest in the previous run first, preventing a slow file from holding the run open after other workers finish.

— Clear mechanism, no flag or config to change behavior. `e2e@0.21.2`

35 **`observationTruncated` step flag** NEW 38

Records `observationTruncated: true` on an agent step when any semantic observation in that step was incomplete.

— Named field, but no consumption guidance given. `e2e@0.21.3`

36 **Select-option filtering excludes hidden options** IMPROVED 35

Explicitly hidden select options are now excluded before applying the observation choice limit, while normal closed-select choices and disabled states are still retained.

— Describes behavior but no API surface to act on. `@e2e-dev/web@0.15.3`

37 **Early `browser.waitForResponse` support** IMPROVED 32

`browser.waitForResponse` can now be called before the first `browser.goto` or `app.open`.

— One-line behavior change, no mechanism given. `@e2e-dev/web@0.15.3`

38 **Native dialog kind exposed in onDialog** IMPROVED 32

`browser.onDialog` handlers now receive the native dialog kind.

— Bare statement of a field addition. `@e2e-dev/web@0.15.3`

39 **session.report() returns hosted Report** IMPROVED 28

After close and a successful trace write, session.report() returns the hosted session's `Report`.

— One-line API behavior, minimal context. `e2e@0.21.2`

40 **ARIA role attribute fallback parsing** IMPROVED 27

Reads the first recognized ARIA role token from a role attribute and retains native semantics when no recognized role is present.

— Thin internal parsing note, no surface to act on. `@e2e-dev/web@0.15.1`

41 **Cached action model reporting** IMPROVED 27

Reports the model behind cached actions and flags replays made with a different model.

— Bare description, no field or command named. `e2e@0.21.2`

42 **e2e mcp server listed in MCP Registry** NEW 25

The `e2e mcp` server is now listed in the official MCP Registry as `io.github.tester-army/e2e`.

— Informational listing, nothing to configure. `e2e@0.21.2`

43 **iOS viewport, keyboard, and text-field reliability fixes** IMPROVED 21

Fixes iOS viewport, keyboard, and text-field reliability issues in `@e2e-dev/mobile`.

— Summary-only mention with no mechanism or scope. `@e2e-dev/mobile@0.12.4`

╰──▷ BREAKING ON UPGRADE

! Third-party engines must handle the new `union` kind of `LocatorExpression` introduced by locator.or(other).

! A project tool can no longer be named `test_lines` — the name is now reserved by the built-in MCP catalog tool.

! device.setNetwork('offline') and device.setAirplaneMode(true) now fail with `UNSUPPORTED_CAPABILITY` on iOS; use `platforms: ['android']` with setAirplaneMode(true) (Android 11+) for offline tests. Note: setNetwork('offline') disables Wi-Fi only and does not cut cellular data.

! device.locator() and `selector` locator role values have changed: `role=textbox` must become `role=text-field`, `role=navigation` must become `role=navigation-bar`, `role=search-field` must become `role=search`, and `role=listitem` must become `role=cell`. Selectors using the old role names will match different or no nodes.

WAS THIS USEFUL? 👍 👎

martin-loop

Feedback ⓘ v0.8.9 NOTES ↗ RANK

Run coding agents without babysitting them. Keep jobs focused, bounded, checked and accountable from start to finish.

> martin-loop's biggest update this window adds explicit provider fallback chains and a `martin continue` subcommand for resuming interrupted runs, alongside breaking default protections that block secret/key file writes and destructive test edits unless explicitly approved. Smaller CLI and MCP improvements cover verifier-command validation, rollback safety across symlinks, workspace-scoped run listing, and badge/status reporting.

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

01 Default blocking of secret and key file writes

BREAKING

95

Blocks writes to secret and key files by default — `.env` and `.env.*` non-templates, `*.pem`, `*.key`, `*.p12`, `*.pfx`, and SSH private keys — stopping the run unless explicitly permitted via `--allow-path`. Extends ignored-secret and denied-path file monitoring to detect edits and deletions including same-size content changes, restoring blocked changes when the original state can be authenticated; the MCP server enforces the same checks on denied or protected ignored files, requires an explicit allowed path for secret-file writes, and accurately reports the resulting rollback outcome. Breaking: existing workflows that let agents modify these files will break unless the paths are listed explicitly with `--allow-path`.

Run an agent task while allowing it to rotate secrets stored in `.env.production` — a path blocked by default.

```
$ martin-loop run --allow-path .env.production 'rotate the API key in .env.production'
```

— Enumerates all protected file patterns, the exemption flag, and MCP enforcement. v0.8.7 mcp-v0.8.7

02 Provider fallback chains for governed runs NEW

93

Adds a repeatable `--fallback-engine` CLI flag to specify an explicit ordered provider fallback chain, with the same control exposed via `fallbackEngines` in MCP's `martin_run`. Fallback providers are validated for availability and authentication before any spend occurs and cannot silently widen to a new vendor; USD, token, iteration, and wall-clock budget plus run authority (task, scope, verifier) are preserved across a

provider switch, fallback stays disabled for external-write/irreversible recovery runs, and the completed run reports the provider that actually executed the work.

Run a governed loop with an explicit ordered provider fallback chain so that spend never silently widens to an unintended vendor.

```
$ martin run --fallback-engine provider-a --fallback-engine
provider-b --fallback-engine provider-c <loopId>
```

Specify an ordered list of fallback providers so a governed run can continue under budget if the primary provider is unavailable.

```
$ npx -y @martinloop/mcp@0.8.9
```

— Names flags, fields, and runnable commands for both CLI and MCP. v0.8.9 mcp-v0.8.9

---

03 **Approval gating for destructive test changes** BREAKING

91

Adds `--approve-test-changes` CLI flag to gate destructive test modifications — deleting tests, removing or changing assertions, dropping cases, or marking cases skipped — while allowing additions, appended cases, and import/formatting edits without approval; swarm children cannot escalate beyond parent authority. The MCP server enforces the same rule, requiring explicit approval to delete an existing test or remove/change its assertions, and Swarm contracts cannot widen test-change approval beyond parent authority. Breaking: runs that previously performed these operations without the flag will now be blocked.

Permit an agent to clean up obsolete tests (deletions, removed assertions) that would otherwise be blocked without explicit approval.

```
$ martin-loop run --approve-test-changes 'remove deprecated
integration tests for the v1 API'
```

— Names flag, exact gated operations, and breaking migration note. v0.8.7 mcp-v0.8.7

---

04 **martin continue resumes interrupted reversible runs** NEW

90

Adds `martin continue <loopId>` subcommand to resume eligible interrupted reversible runs, reusing the original loop ID, task, verifier, scope, budget, and evidence. USD, token, iteration, and wall-clock limits remain bound to the original run rather than

resetting, and terminal, exhausted, uncertain-active, irreversible, and external-write runs fail closed instead of being automatically replayed.

Resume an interrupted reversible run mid-repair without losing accumulated evidence, budget, or verifier state.

```
$ martin continue <loopId>
```

— Exact subcommand with clear resume semantics and fail-closed cases. v0.8.9

05 **Shields-compatible fields in governed badge JSON** IMPROVED 80

Adds Shields endpoint fields (`schemaVersion` , `label` , `message` , `color`) to the `badge --governed --format json` response alongside the existing `command` , `governed` , and `markdown` fields.

Retrieve a governed badge as JSON with Shields.io-compatible fields for embedding in a status dashboard.

```
$ martin-loop badge --governed --format json
```

— Enumerates every new and existing field in the JSON response. v0.8.5

06 **Verifier command validation for malformed quotes and flags**

IMPROVED 78

`martin_preflight` checks for malformed verifier quotes and MartinLoop-owned flags inside verifier commands, refusing to mark malformed commands as ready for execution; `run` , `preflight` , and `enable` now reject a `--verify` value containing an unterminated quote or martin-loop's own options with `invalid_input` and exit code 2.

— Names exact flag, error code, and subcommands affected. mcp-v0.8.7 v0.8.5

07 **Workspace-wide run listing and scoping** NEW 75

Adds `--all-workspaces` flag to `runs list` to query the entire runs root across all workspaces, not just the current one; `runs list` and `runs get --latest` now report the scoped workspace, and explain how many runs exist in other workspaces when the current workspace has none.

List every run across all workspaces in the runs root — useful when the current workspace shows no runs but others do.

```
$ martin-loop runs list --all-workspaces
```

– Exact flag and scoping behaviour with a runnable example. v0.8.5

THINNER COVERAGE BELOW

08 **Argument schema completeness across MCP tools** IMPROVED

50

`martin_dossier`, `martin_pr_summary`, `martin_create_pr`, `martin_review_pr`, `martin_pause`, `martin_cancel`, and `martin_continue` now accept every argument they advertise.

– Names all affected MCP tools but describes only a schema fix. mcp-v0.8.6

09 **Blocking failed Claude Code sessions from verification** IMPROVED

45

Blocks failed Claude Code sessions, including terminal `is_error` results, from being passed to a green verifier and reported as successfully completed.

– Names the failure condition but no reproduction path. v0.8.7

10 **Verified-run gating for PR creation** IMPROVED

45

`martin_create_pr` with `execute: true` opens a pull request only for a verified run with intact receipts.

– Names the tool, flag, and gating condition briefly. mcp-v0.8.6

11 **Fail-closed rollback across symlinks and junctions** IMPROVED

40

Rollback across symlinks or Windows junctions now fails closed instead of deleting files outside the repository.

– States the behaviour change but no mechanism or reproduction steps. mcp-v0.8.7

12 **Codex diagnostics in preflight checks** NEW

40

Adds Codex diagnostics to the advertised output schema of `martin_preflight`, returned on hosts where Codex is installed.

– Names the tool and condition but not the diagnostic content. mcp-v0.8.6

13 **NEEDS_REVIEW status in arcade status** NEW

35

`martin_arcade_status` now reports runs whose outcome is `NEEDS_REVIEW`.

– Names the tool and new status value, nothing further. mcp-v0.8.6

14 **Warning for no-op edit runs** NEW

30

Flags edit runs that change no files with a warning in the verified handoff.

– Single-line description with no further detail given. v0.8.7

└──▷ ALSO FROM THESE RELEASES

Pin and start the versioned MCP package to get the fail-closed preflight and rollback hardening in this release.

```
$ npx -y @martinloop/mcp@0.8.7
```

└──▷ BREAKING ON UPGRADE

- ! Writes to `.env`, `.env.*` (non-templates), `*.pem`, `*.key`, `*.p12`, `*.pfx`, and SSH private key files now stop the run by default; existing workflows that allow agents to modify these files will break unless the paths are listed explicitly with `--allow-path`.
- ! Deleting a test, removing or changing its assertions, dropping a case, or marking one skipped now requires `--approve-test-changes`; runs that previously performed these operations without the flag will be blocked.

WAS THIS USEFUL? 👍 👎

Superset

Feedback ⓘ cli-v1.40.0 NOTES ↗ RANK

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

> Superset shipped end-to-end ACP agent sessions spanning host, desktop and a redesigned mobile app, alongside new CLI workspace/wake commands, a Mistral Vibe agent, terminal connection diagnostics, and desktop infrastructure work including memguard OOM protection and a Durable Objects-backed presence system.

└──▷ **WHAT SHIPPED · 18 FEATURES** most completely described first ? what's the number?

01 **CLI `workspaces get` and host wake commands** NEW 80

Adds a `workspaces get` subcommand and configurable per-host `set-wake` and `wake` commands to the CLI, letting you fetch workspace details or trigger a host's wake command without the desktop UI.

Fetch details for a specific workspace by ID without opening the desktop UI.

```
$ superset workspaces get <workspace-id>
```

Manually trigger the configured wake command for a host that is currently offline.

```
$ superset wake <host-id>
```

— Runnable commands shown directly cli-v1.40.0

02 **DESKTOP_DISPLAY_NAME env var and close-warning setting** NEW

75

Adds a `DESKTOP_DISPLAY_NAME` environment variable to control the desktop app's display name, and a setting to restore the running-terminal close warning in the desktop app.

Override the desktop app's display name in a custom deployment without rebuilding the binary.

```
$ DESKTOP_DISPLAY_NAME='Acme Dev' open
/Applications/Superset.app
```

— Named env var with a runnable override example `desktop-v1.40.0`

03 **ACP agent sessions across host, desktop and mobile** NEW

70

Adds live ACP agent sessions end-to-end (host harness, mobile UI, desktop gate) plus ACP-backed workspace pages on mobile — a restyled home, native workspace screen and diff surface — and an ACP sessions API + CLI for desktop.

— Names surfaces across platforms but no exact commands shown `desktop-v1.40.0` `cli-v1.40.0`

04 **Mobile app UI and session overhaul** NEW

65

Adds live chat sessions over the relay on mobile, host-owned workspace data, single-stack navigation, settings, and a new-chat widget, plus a unified home, attachments sheet, and glass composer with voice dictation. Later adds on-device 'dictate as you go' with live dictation visible in the collapsed composer, and swipe-in from the right edge to return to the last workspace.

— Many named mobile features but no commands or settings paths `cli-v1.40.0` `desktop-v1.40.0`

05 **Terminal connection diagnostics and close safeguard** IMPROVED

65

Adds a terminal connection status indicator with a diagnosis popover, a confirmation prompt before closing a terminal pane with a running process, and surfaces why a terminal cannot connect along with failure telemetry in the desktop.

— Clear UI mechanism but no exact navigation path `cli-v1.40.0`

06 **`eas-dev.sh --user` dev sign-in flag** NEW

65

Adds an `eas-dev.sh --user` flag to sign the iOS simulator app in as a dev account.

Sign the iOS simulator into a dev account so you can test auth-gated mobile features locally.

```
$ bash eas-dev.sh --user
```

— Exact flag with runnable command `desktop-v1.40.0`

07 **Sandbox bundle pinning for environments** NEW 65

Adds a `sandbox_bundles` table to record every released bundle, seeded with the bundle the release last pinned, and adds `environments.bundle_sha` as a NOT NULL column ensuring every environment and box carries a bundle pin.

— Named schema change but no user-facing action `desktop-v1.40.0`

THINNER COVERAGE BELOW

08 **`pdev` subagents as steerable child sessions** NEW 55

Adds `pdev` subagents as steerable child sessions with a single drawer for running work, shipped as pdev 0.2.0.

— Names version and concept but no usage path `desktop-v1.40.0`

09 **Desktop chat pane UI improvements** NEW 50

Adds a center pane area for terminals and chats, a `/mcp` picker, and the ability to edit a sent message in the desktop app.

— Named UI additions but no detail on mechanism `desktop-v1.40.0`

10 **memguard OOM protection** NEW 45

Adds memguard to rank processes for the OOM killer and reap memory before the kernel does.

— Mechanism explained but no configuration surface given `desktop-v1.40.0`

11 **Presence moved to Durable Objects** IMPROVED 45

Moves presence from Postgres to Durable Objects, keeping page and workspace visit history in realtime hubs.

— Before/after given but no user-facing control `desktop-v1.40.0`

12 **Linear project and cycle filtering in Tasks** IMPROVED 40

Adds Linear project and cycle filtering and sorting to the Tasks view.

— Names the view but no exact controls `cli-v1.40.0`

13 **`cmd+f` search in v2 changes pane** NEW 40

Adds `cmd+f` search to the v2 changes pane in the desktop.

— Named shortcut but thin on scope `cli-v1.40.0`

---

14 **Mistral Vibe coding agent** NEW — 35

Adds first-class Mistral Vibe coding agent to the desktop.

— Names the agent but no setup details `cli-v1.40.0`

---

15 **GPT-5.6 models in Codex model picker** NEW — 35

Adds GPT-5.6 models to the Codex model picker.

— Names the model but no further detail `cli-v1.40.0`

---

16 **Host deletion for owners** NEW — 35

Allows owners to delete hosts from the desktop.

— Clear permission and action, minimal mechanism `cli-v1.40.0`

---

17 **Workspace hub access enforcement** NEW — 30

Adds workspace hub enforcement of who may open a workspace via the realtime hub.

— Thin description, no mechanism or scope detail `desktop-v1.40.0`

---

18 **Sydney relay region** NEW — 30

Adds a Sydney (`syd`) region to the relay fleet.

— Names region code but no configuration path `cli-v1.40.0`

WAS THIS USEFUL?  

Block Goose

Feedback ⓘ v1.56.0 NOTES ↗ CODE ↗ RANK

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

> Goose v1.56.0 adds a concurrency cap for subagents, shell-history context for @goose, support for the IO Intelligence provider, OSC 7501 terminal session-status reporting, and a Windows NSIS installer, alongside a new Boa JavaScript runtime replacing PCTX code mode and resumable, UI-visible subagent sessions.

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

01 Shell history context flag for term init NEW

78

Adds the `--dangerous-include-history` flag to `goose term init` so recent shell commands are recorded as context for `@goose`; the release notes warn command lines may contain credentials that get sent to the model provider.

– Exact flag and subcommand named with a safety caveat, no example shown. v1.56.0

02 Subagent management enhancements NEW

77

Adds the `GOOSE_MAX_CONCURRENT_SUBAGENTS` environment variable to cap the number of subagents running in parallel, enables resuming failed or cancelled delegate subagents, and adds subagent session links in the UI with view-only subagent session access.

Prevent runaway parallelism in multi-agent workflows by capping how many subagents can run at once.

```
$ export GOOSE_MAX_CONCURRENT_SUBAGENTS=4
```

– Names exact env var and includes runnable export example. States capability but no command or flag given. Names UI surface but no exact navigation path given.

v1.56.0

THINNER COVERAGE BELOW

03 Boa JavaScript runtime replaces PCTX code mode BREAKING

50

Replaces the PCTX code mode with a Boa JavaScript runtime for in-agent script execution.

– Names old and new mechanisms but no usage example. v1.56.0

04 OSC 7501 session status reporting NEW

35

The CLI now reports session status to terminal emulators via OSC 7501 escape sequences.

— Names the mechanism but no usage detail given. v1.56.0

---

05 **Provider-defined cache pricing for cost reporting** IMPROVED 25

Supports provider-defined cache prices for more accurate cost reporting.

— Describes benefit without naming config or command. v1.56.0

---

06 **Windows NSIS installer** NEW 25

Adds a Windows NSIS installer for Goose.

— Named but no install steps or path provided. v1.56.0

---

07 **IO Intelligence model provider support** NEW 20

Adds support for the IO Intelligence provider.

— Only mentioned by name in the summary, no further detail. v1.56.0

---

08 **Automatic thinking-level operation** IMPROVED 15

Adds automatic thinking-level operation for improved model reasoning.

— Only a bare description, no mechanism or config named. v1.56.0

WAS THIS USEFUL?  

Charm Crush

Feedback ⓘ v0.98.2 NOTES ↗ CODE ↗ RANK

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

> Crush v0.98.2 ships a new built-in theme, OpenRouter request attribution for sticky routing, and signed release manifests for artifact verification.

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

01 **OpenRouter app attribution with session_id** NEW 50

Crush now sends app attribution headers and a `session_id` to OpenRouter, enabling sticky routing of requests tied to a session.

— Names the mechanism but no config/flag to invoke it v0.98.2

02 **Signed Packslip release manifests** NEW 40

Crush publishes optional signed Packslip release manifests, allowing artifact verification of releases.

— Describes purpose but not how to verify or enable v0.98.2

03 **Built-in keanu theme** NEW 35

Crush adds a new built-in `keanu` theme.

— Named theme but no setup detail beyond its existence v0.98.2

WAS THIS USEFUL? 👍 👎

All Hands AI OpenHands

Feedback ⓘ v1.33.0 NOTES ↗

RANK

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

> OpenHands v1.33.0 adds new launch flags for its verification tool and makes DigitalOcean web sessions persist across restarts.

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

01 Cloud-lock and web UI flags for verify-openhands NEW

75

The `verify-openhands` tool gains `--mars-web` and `--lock-to-cloud` launch flags, letting a user run it locked to cloud infrastructure with the Mars web interface enabled, e.g. `verify-openhands --mars-web --lock-to-cloud`.

Launch verify-openhands locked to cloud infrastructure with the Mars web interface enabled.

```
$ verify-openhands --mars-web --lock-to-cloud
```

— Named flags plus a runnable command, but little mechanism detail. v1.33.0

THINNER COVERAGE BELOW

02 Persistent sign-in for DigitalOcean web build IMPROVED

35

The DigitalOcean web build now stays signed in across restarts, eliminating the need for repeated authentication.

— Describes the behavior change but no config, flag, or steps to use it. v1.33.0

WAS THIS USEFUL? 👍 👎

juggler

Feedback ⓘ v0.7.8 NOTES ↗ RANK

The Juggler Code Agent

> Juggler's v0.7.7 and v0.7.8 releases add a new `run_code` tool for full code mode execution, a `decide` extension op that lets Auto-approve judge tool calls via a decision model, remote server connections, global MCP pre-approval, conversation export to HTML/Markdown, and a wide set of smaller extension SDK, networking, and UI refinements.

└──▷ **WHAT SHIPPED · 18 FEATURES** most completely described first ? what's the number?

01 **`--allow-host` flag for reverse-proxy serving** NEW 80

Adds a repeatable `--allow-host <name>` flag to serve Juggler behind a reverse proxy such as tailscale serve, counting requests under that name as remote viewers.

Expose a long-running Juggler session through tailscale serve so team members can reach it by hostname without opening a raw port.

```
$ juggler --allow-host mymac.local --allow-host juggler
```

— Exact flag, example command, and remote-viewer accounting behavior. v0.7.7

02 **Extension API and SDK additions** NEW 75

Adds a new `decide` extension op that asks a decision model yes/no, choice, and score questions, with Auto-approve now able to judge tool calls using a decision model chosen in Settings, and adds Extension SDK helper functions `formatTokens`, menu-keyboard helpers, live-task-output helpers, and `hasStrategy`.

Validate and link an extension that uses the new `decide` op to gate auto-approve decisions via a decision model.

```
$ juggler ext validate ./my-extension && juggler ext link ./my-extension
```

— Explains mechanism and Settings integration but no direct command. Names the new helpers but gives no usage detail.

v0.7.8

03 **Conversation export as HTML and Markdown** NEW 75

Adds the ability to export a conversation as a self-contained HTML page (via its tab menu or the `/export` route) and to copy a whole conversation as Markdown (via its tab menu

or footer).

Share a finished debugging session with a colleague who doesn't have Juggler installed — export it as a portable HTML file from the conversation's tab menu or navigate directly to the export route in your browser.

```
$ open http://localhost:<port>/export
```

— Names the exact route and UI paths with a runnable example. v0.7.7

04 **Extension op cancellation and camelCase IDs** IMPROVED 70

Extension ops `stat`, `mkdir`, `getTree`, `expandDirectory`, and `fileHash` now accept an `AbortSignal` for cancellation, and extension shell ops now accept camelCase ids — `taskId`, `shellId`, `convId` — with snake_case still supported.

— Names every affected op and id but no runnable example. v0.7.8

05 **`run_code` tool and `image()` function** NEW 70

Adds a new `run_code` tool, a mutating counterpart to `query_code` that unlocks full code mode features, along with an image() function usable inside both `run_code` and `query_code` programs so the model can receive images from a script.

— Names the tools and function with mechanism but no runnable example. v0.7.7

06 **Package-manager-delegated update management** NEW 65

Packagers can set `JUGGLER_UPDATES_MANAGED_BY` to delegate update management to their own package manager instead of Juggler's built-in updater.

Let your Linux distro's package manager own Juggler updates instead of the built-in updater.

```
$ JUGGLER_UPDATES_MANAGED_BY=apt juggler
```

— Exact env var and usage shown, but no further mechanism detail. v0.7.8

THINNER COVERAGE BELOW

07 **Max output limit increase and retry workflow** IMPROVED 55

Models reporting no output limit now get a Max output up to 64k on large windows; a reply cut off at Max output offers a one-off retry with double the limit, and while a reply streams the status bar shows token count against Max output.

— Gives concrete 64k limit and retry behavior but no config surface. v0.7.8

08 **Global MCP server pre-approval** NEW 55

Pre-approve a global MCP server in Settings so its tools never prompt for permission; Don't ask again on an MCP tool now covers every conversation in the project.

— Explains scope change but only a Settings path to act on. v0.7.7

09 **Remote server connections** NEW 50

Session ▸ Connect to Server… opens a live window onto another machine's Juggler server, reconnected automatically at launch.

— Names the menu path and reconnect behavior, no further mechanism. v0.7.8

10 **Configurable sub-agent run budget** NEW 50

Sub-agent run budget is now configurable in Settings → Defaults, defaulting to 1 M tokens.

— Names the setting path and default value. v0.7.7

11 **Git pin scope expansion** IMPROVED 50

Git pin now reviews any scope: staged, unstaged, branch, last commit, or a git diff range.

— Lists the scopes covered but no command or path given. v0.7.7

12 **Simplified Chinese language support** NEW 40

Adds Simplified Chinese (`zh-CN`) translation, selectable under Settings → Language.

— Names the locale and setting path; nothing more to describe. v0.7.8

13 **WebSearch auto-retry on rate limiting** IMPROVED 35

WebSearch retries automatically when DuckDuckGo rate-limits it and surfaces a notice if the delay persists.

— Describes behavior but no configuration or command. v0.7.8

14 **Sidebar plan progress card** NEW 35

Adds a sidebar progress card showing plan and to-do progress, clickable to open in the Pinboard.

— Describes the UI element but no further detail. v0.7.7

15 **batch_read properties panel actions** IMPROVED 35

`batch_read` 's properties panel lists files with open, copy, reveal, and pin buttons.

— Names the tool and buttons but no further mechanism. v0.7.7

16 **LAN hostname resolution for servers** IMPROVED 30

LAN servers respond to local hostnames such as `juggler` or `mymac.local` in addition to IP addresses.

— States the behavior with example hostnames, no setup steps. v0.7.7

17 **Per-machine theme and zoom settings** IMPROVED 25

Theme and zoom settings are now per machine and survive reconnecting to a different server.

— Brief behavior description, no mechanism or setting path given. v0.7.8

18 **Conversation tip and prompt toggles** IMPROVED 25

New conversations' tips can be turned off; Suggest prompts now covers starter prompts.

— Minimal description of a settings toggle. v0.7.7

WAS THIS USEFUL? 👍 👎

harness-remote

Feedback ⓘ v3.5.0 NOTES ↗ RANK

Native-session control plane for Codex, Claude Code, OpenCode, OMP and PI. Run, resume and hand off coding sessions across your machines.

> Harness Remote shipped mid-turn prompt queuing and background subagent auto-resume for Claude Code, LAN phone pairing for the mobile companion app, a redesigned per-reply agent Activity view, Claude reasoning-effort controls, and Brazilian Portuguese localization, alongside a string of session-reliability and UI fixes.

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

01 **LAN phone pairing for mobile companion app** NEW 80

Adds a Phone pairing setting in the desktop app that opens its built-in daemon to the local network (off by default, requires confirmation and a runtime restart), plus a Pair a phone action on every machine card that displays a live QR code with countdown from that machine's daemon until the phone is paired.

Pair a mobile device to a specific remote machine over LAN without leaving the Machines list.

📍 In the Machines list, click the card for the target machine, select Pair a phone, then scan the displayed QR code with the Harness Remote app on your phone before the countdown expires.

Enable LAN phone pairing on the desktop app's local daemon so mobile devices on the same network can connect.

📍 In the desktop app, go to Phone pairing settings, enable Open to local network, confirm the prompt, and allow the local runtime to restart.

— Exact settings path and step-by-step pairing flow given v3.3.0

02 **Conversation activity visibility settings** NEW 80

Adds Settings → Conversation activity to control what stays visible while the agent works — choose from latest update, all updates, every step, or a summary line — and whether reasoning is collapsed, expanded, or hidden; changes apply to open Sessions immediately.

Control agent verbosity mid-session without restarting — set activity detail and reasoning visibility to match your workflow.

📍 In the desktop app, go to Settings → Conversation activity, set the while agent works display to every step and reasoning to expanded, then return to any open

Session to see changes take effect immediately.

— Named setting path with full option list and example v3.3.0

03 **Mid-turn prompt queuing and steering** NEW 75

Prompts sent mid-turn can now queue — shown in a Queued strip above the composer — or steer directly into the live turn (Codex harness), with per-prompt × withdrawal and queue state persisted in the daemon across Session switches and app reloads.

— Names exact UI elements and persistence behavior v3.4.0

04 **Background subagent auto-resume for Claude Code** NEW 70

Background subagents, `run_in_background` commands, and monitors launched by Claude Code now automatically resume their Session, delivering the follow-up turn as a new reply without any user prompt, and closing after 30 seconds of silence.

— Mechanism and exact trigger command named v3.4.0

05 **Redesigned per-reply agent Activity section** IMPROVED 70

Redesigns the agent activity display into a collapsible Activity section per reply, showing a summary line (duration, command/edit/read/search counts), compact tool-call rows, and a Thought for Ns reasoning line with preview.

— Describes layout and content but no navigation steps v3.3.0

06 **Live cross-client session status updates** IMPROVED 60

Harness Remote now shows a live busy state in the Session list for Codex and Claude Code sessions launched from a terminal or desktop client while their turns are running, and updates the Session list immediately when OMP, PI, and Claude Code turns finish — without waiting for the next 30-second poll or reopening the Session.

— Names clients and poll interval but behavior is automatic v3.5.0

07 **CODEX_HOME detection for bundled Codex executable** IMPROVED 60

Harness Remote now reads `CODEX_HOME` to detect the bundled Codex executable even when its PATH entry is a dangling symlink, reusing its cached account without forcing a browser login.

— Env var and failure mode named, but no user action required v3.4.0

THINNER COVERAGE BELOW

08 **Reasoning-effort levels for Claude adapter** NEW 55

The model picker for the installed Claude adapter now exposes reasoning-effort levels, including a Default option and model-specific levels, letting users choose a model's reasoning effort directly from the picker.

— UI location named but exact levels not listed v3.5.0

09 **Session management for OpenCode 2, MiMo and Copilot**

NEW INTEGRATION IMPROVED 50

Supports renaming and removing OpenCode 2, MiMo, and Copilot Sessions from Harness Remote again, and loads OpenCode 2 history even when the Session's project folder has been deleted.

— Names three affected clients and two restored behaviors v3.3.0

10 **Project folder configuration in machine editor** IMPROVED

50

Adds project folder configuration in the machine editor; the desktop app's own daemon now defaults to the home folder instead of its launch directory.

— UI location named, mechanism of default change described v3.3.0

11 **Bounded paging for session discovery** IMPROVED 45

Session discovery now pages through empty project roots (bounded to four requests) so Sessions in later project roots are found automatically on startup.

— Concrete bound given but no user-facing action v3.4.0

12 **Right-click context menu for text fields and links** NEW

45

Adds a right-click context menu with Cut, Copy, Paste, and Select all in text fields, Copy on selected text, and Copy link address on links.

— Lists menu items but no deeper mechanism v3.3.0

13 **Independent machine loading on timeout** IMPROVED 35

Available machines and their Sessions keep loading independently when a saved machine cannot be reached, instead of blocking on that machine's timeout.

— Describes before/after but no concrete numbers or path v3.5.0

14 **Cleaned-up Codex desktop message display** IMPROVED 35

Codex desktop-app messages are now displayed without surrounding UI context markup in both saved history and live replay.

— Thin UI cleanup with no further detail v3.5.0

15 **Brazilian Portuguese (pt-BR) localization** NEW 35

Adds complete Brazilian Portuguese (pt-BR) translations, with common Portuguese browser locale variants mapped automatically to pt-BR.

— Names locale code but little mechanism beyond mapping v3.5.0

NEW INTEGRATION GitHub Copilot CLI integration

NEW INTEGRATION MiMo Code native ACP integration

UPDATED SUPPORTED PLATFORM Desktop platform builds

v3.2.0 NOTES ↗ STABLE

Harness Remote 3.2.0 adds a `--verbose` gateway flag, a Ready marker for finished Sessions, Desktop focus mode, and a compact Session header.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v3.2.0 https://github.com/giuliastro/harness-remote.git
# already have the repo? check out this version:
$ git checkout v3.2.0
```

Adds `--verbose` flag to the gateway for full adapter diagnostics beyond harness-only terminal output.

Adds a Ready marker that keeps a finished Session highlighted until the user opens it, cleared on next click, key press, scroll, or mouse movement.

Adds Desktop focus mode that hides the Session header and toolbar so the transcript fills the full pane.

Adds a compact single-row Session header on wide panes, with project outcome as a chip popover and sent messages lifted to the top of the transcript.

Opening a linked Session on another harness now requires an explicit Harness choice, preventing duplicate Claude or MiMo Sessions on a normal Send.

+1 more

WAS THIS USEFUL? 👍 👎

JuliusBrussee Caveman

Feedback ⓘ bin-v4.0.2 NOTES ↗

RANK

why use many token when few token do trick. Viral skill + proxy for coding agents that cuts 65% of tokens by talking like a caveman.

> Caveman extended its token-compressing proxy/engine binaries to cover six more languages beyond Go and started shipping signed cross-platform runtime builds.

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

01 **Multi-language code compression in proxy/engine binaries**

IMPROVED 62

The `caveman-proxy` and `caveman-engine` binaries now compress TypeScript, JavaScript, Python, Rust, Java, and C/C++ in addition to Go; the `caveman-proxy` container image itself remains Go-only.

— Names binaries and languages but no flags or usage shown bin-v4.0.2

THINNER COVERAGE BELOW

02 **Signed runtime binaries for macOS, Linux, Windows** NEW

38

Caveman now releases signed runtime companion binaries for macOS, Linux, and Windows, distributed under the Apache License 2.0.

— States platforms and license but no install or usage steps bin-v4.0.2

WAS THIS USEFUL? 👍 👎

Amazon Kiro

Feedback ⓘ

Models: Claude Haiku 5.5 Now Available (2026-10-23) BLOG ↗ RANK

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

> Kiro's biggest update this window is persistent Memory that now carries repository conventions and context across chat sessions in both the CLI and IDE, alongside a new workflow-delegation frequency control and a set of V3 agent safety and CLI usability additions.

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

01 Persistent memory across chat sessions (CLI and IDE)

NEW SUPPORTED PLATFORM NEW 91

Kiro now carries context such as repository conventions, corrections, and ruled-out approaches between sessions. In the IDE this is controlled by the `Kiro Agent: Memory` setting, including a `Use only` mode that lets Kiro read memories without modifying them, a new Memory view accessible via the Memories icon in the activity bar, and chat surfacing when Kiro accesses or changes a memory entry. In the CLI, memories are managed via `/memories`, where selecting an entry and pressing `Ctrl+D` deletes it.

Prevent Kiro from writing new memories while still letting it draw on existing ones — useful when you want a stable, curated memory set.

> 📍 In the IDE, open Settings and set `Kiro Agent: Memory` to `Use only`. Kiro will read existing memories each session but will not add or update any entries.

Remove a stale or incorrect memory entry from the CLI without leaving the terminal.

```
$ /memories  # then arrow-key to the target memory and press Ctrl+D
```

— Names exact setting, mode, UI path and CLI command/shortcut.

IDE 1.4.26: Memory Across Chat Sessions… CLI: MCP Elicitation, Workflow Controls…

02 Workflow delegation frequency control NEW 70

A new `Workflow use` (CLI) / `Use workflows` (IDE) setting adds options `Only when asked`, `Only for large tasks`, and `As much as possible` to control when Kiro starts Workflows. The existing `Kiro Agent: Workflows` setting remains the master on/off switch, independent of this new frequency control.

Set workflow delegation frequency to hand off only large, multi-step tasks while keeping simple requests direct.

> 📍 In a chat, open the Workflows panel › set `Use workflows` to `Only for large tasks`.

— Names setting keys and options with exact UI steps.

IDE: Choose When Kiro Uses Workflows (2… | CLI: MCP Elicitation, Workflow Controls…

---

03 **`--trust-workspace` flag for V3 CLI start** NEW 70

Adds `--trust-workspace` flag to the V3 CLI start command to skip the workspace trust prompt for a single launch, without saving the choice.

Launch a V3 session in a CI or one-off environment where you want to skip the trust prompt without permanently trusting the workspace.

```
$ kiro --trust-workspace
```

— Exact flag and runnable command given. CLI: MCP Elicitation, Workflow Controls…

---

04 **`/model edit` command to hide unused models** NEW 60

Adds `/model edit` command to hide unused models from the picker; hidden models remain available by ID.

Clean up your model picker so only the models you regularly use appear, reducing noise during model selection.

```
$ /model edit
```

— Exact command named and runnable. CLI: MCP Elicitation, Workflow Controls…

---

05 **V3 approval dialog changes** IMPROVED 60

Improves V3 approval dialogs: unparsable shell commands are now denied outright instead of prompting for approval, and dialogs list `Allow once` and `Deny once` first, followed by choices that save a persistent allow or deny rule.

— Mechanism and ordering named, but no runnable example.

CLI: MCP Elicitation, Workflow Controls…

THINNER COVERAGE BELOW

---

06 **MCP elicitation in local V3 sessions** NEW 55

Enables MCP elicitation in local V3 sessions: MCP tools can now pause and show a form or link when they need more information, then continue without restarting the call.

— Mechanism explained but no concrete command to try it.

CLI: MCP Elicitation, Workflow Controls…

07 **Inline workflow plan editing** NEW 50

Adds inline workflow plan editing: type suggested changes directly into a V3 workflow plan approval before any steps run.

— Describes behavior but no navigation or command given.

CLI: MCP Elicitation, Workflow Controls…

08 **Tighter sub-agent MCP boundaries** IMPROVED 45

V3 sub-agents no longer receive MCP servers declared inline by their parent agent, tightening sub-agent MCP boundaries.

— Clear before/after but no actionable step for the reader.

CLI: MCP Elicitation, Workflow Controls…

09 **Confirmation prompt before deleting V3 session** NEW 25

Adds a confirmation prompt to the V3 session dashboard before deleting a session.

— Bare description, no further detail or usage path.

CLI: MCP Elicitation, Workflow Controls…

UPDATED SUPPORTED PLATFORM Claude Fable 5.1, Claude Sonnet 5.5, GPT-5.6 Terra, and GPT-5.6 Luna — General Availability

> Models: Claude Haiku 5.5 Now Available (2026-10-23) BLOG ↗ LATEST
>
> STABLE
>
> Claude Haiku 5.5 brings a 1M-token context window to Kiro, with tiered credit multipliers (0.06x up to 100K tokens, 0.3x beyond).
>
> Adds Claude Haiku 5.5 with experimental support across Kiro IDE, CLI, Crew, and Web for Pro, Pro+, Pro Max, and Power customers in `us-east-1` (IAD) and `eu-central-1` (FRA), accessible via `kirocrew restart` for Crew or IDE/CLI restart.
>
> Supports a 1M-token context window, with requests up to 100K tokens billed at a 0.06x credit multiplier and requests exceeding 100K tokens billed at 0.3x.

> Models: Generally Available: Claude Fable 5.1, Claude Sonnet 5.5, GPT-5.6 Terra, and
>
> BLOG ↗ STABLE

> Removes the Kiro Excluded Models indemnity exception for the newly GA models; `kirocrew restart` reloads the updated model list.
>
> Removes the Kiro Excluded Models indemnity exception for Claude Fable 5.1, Claude Sonnet 5.5, GPT-5.6 Terra, and GPT-5.6 Luna.
>
> Loads the updated model list via `kirocrew restart` for Crew, IDE/CLI restart, or browser refresh for Web.

WAS THIS USEFUL? 👍 👎

SST OpenCode

Feedback ⓘ v2.1.4 NOTES ↗ RANK

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

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

01 **Background compaction for session summaries** NEW 85

Adds the `experimental.background_compaction` config key — set to `true` to pre-generate session summaries in the background before the compaction threshold is hit, reducing automatic compaction wait time.

Enable background compaction so sessions near their token limit don't stall waiting for a summary to be generated at compaction time.

yaml

```yaml
experimental:
  background_compaction: true
```

— Exact config key plus runnable YAML example given. v2.1.4

02 **MCP input forms during active tool calls** IMPROVED 70

MCP input forms now surface during active tool calls instead of silently waiting until timeout, supporting text, number, choice, multiselect, boolean, and URL input types across desktop and web.

— Describes behavior and types but gives no exact UI path. v2.1.4

└──▷ BREAKING ON UPGRADE

! Removed endpoint POST `/api` /experimental/voice/session/{sessionID} `/instructions`

WAS THIS USEFUL? 👍 👎

Letta Code

Feedback ⓘ v0.34.21 NOTES ↗ RANK

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

> Letta Code hardened its MCP connection handling, propagating originator tokens through tool calls and isolating originator context across queued turns.

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

01 **MCP originator token propagation and isolation** IMPROVED

47

Letta Code now carries the MCP originator token through to tool-run calls so user-scoped MCP connections authenticate correctly end-to-end. Originators are isolated across queued turns so back-to-back requests don't bleed session context between callers, and originator isolation is advertised on tool requests so downstream MCP handlers can inspect and enforce per-originator boundaries.

— Explains mechanism and scope but names no flags, configs or APIs. v0.34.21

WAS THIS USEFUL? 👍 👎

swttch

Feedback ⓘ v0.34.7 NOTES ↗ RANK

Claude Code GUI plugin for JetBrains IDEs. (ex - Claude Code with GUI)

> swttch's biggest changes this window are a queue editor for steering follow-up messages, a new desktop embed mode (`is_desktop=1`) with URL-controlled top bar visibility, and file attachment support for any file type, alongside smaller additions like background CLI auto-updates, a `Cmd+/` help modal, and session list labeling improvements.

└──▷ **WHAT SHIPPED · 18 FEATURES** most completely described first ? what's the number?

01 Desktop embed mode and top bar visibility via URL parameters

NEW 80

Setting `is_desktop=1` hides the top bar and centres the composer for embedded desktop deployments, exposes applicable webview shortcuts in the desktop help panel, and delegates desktop tab actions to the existing chat controls. A separate URL parameter was also added to hide the top bar outside of full desktop embed mode.

Embed swttch in a desktop shell without the top navigation bar, centred composer layout.

env

```
is_desktop=1
```

— Named config value with mechanism and runnable example. v0.34.2 v0.34.1

02 Cmd+/ command help modal NEW

65

Pressing `Cmd+/` opens a help modal with a quick reference to available commands.

Get a quick overview of available slash commands without leaving the chat panel.

> 📍 In the swttch chat panel, press Cmd+/ to open the help modal listing available commands.

— Exact keyboard shortcut plus runnable usage example. v0.33.3

03 Allow-all-commands approval settings NEW

60

Allow all commands now stays enabled across every session for sponsors, and a separate Allow all in this session option lets any user approve all commands for the duration of a single session.

— Names both approval options with their scope. v0.33.3 v0.34.1

THINNER COVERAGE BELOW

04 **File attachment support for any file type** NEW — 55

Any file type can now be attached to messages, with icons, previews, and the ability to reorder attachments before sending.

— Explains scope and mechanism but no exact UI path. v0.34.1

05 **Context badge token counts and model Effort** IMPROVED — 55

The context badge now displays used/total token counts and an add menu, and the model panel now shows Effort.

— Names specific UI elements added to the badge and panel. v0.34.0

06 **Delete folder-less project from badge** NEW — 50

Folder-less projects can now be deleted directly from the badge shown next to their name in the project list, removing the need to navigate elsewhere to delete them.

Remove a folder-less project without navigating away — use the badge directly next to the project name.

> 📍 In the project list, locate the folder-less project, then click the badge next to its name and select Delete.

— Describes behaviour and exact UI path via example. v0.34.7

07 **/btw floating side-question window** IMPROVED — 50

`/btw` side questions now open in a dedicated floating By the way window instead of appearing inline in the chat.

— Named command with clear before/after but no steps. v0.34.2

08 **spinnerVerbs setting for spinner text** NEW — 50

A new settings row for `spinnerVerbs` lets users control the spinner text shown in the UI.

— Named config key gives a clear starting point. v0.34.2

09 **Edit card line numbers and open-at-line** IMPROVED — 50

Edit cards now display line numbers and clicking them opens the referenced file at the exact line.

— Clear before/after behaviour with concrete UI target. v0.33.4

10 **Keyboard layout and input improvements** IMPROVED — 50

Slash commands can now be typed using any keyboard layout, including Korean layout support, and macOS gains Control-key editing shortcuts along with prompt Undo

support.

– Names platforms and input types but no exact keys. v0.33.3

11 **Background CLI auto-updates and announcements** NEW 50

The Claude Code CLI now updates automatically in the background, mirroring how the interactive CLI handles updates, and swttch announces unread CLI updates and checks for them when auto-updates are enabled.

– Explains mechanism and conditions for the update checks. v0.33.1

12 **Session naming and labeling improvements** IMPROVED 45

Session titles now match Claude Code's own clients, aligning naming conventions with the official CLI; the session list now shows `/rename` names for easier identification; and the panel loading label shows (SSH) when running under Remote Development environments.

– Thin description with no mechanism detail. Names both label changes and their trigger conditions.

v0.34.6 v0.33.3

13 **Stalled-screen recovery guide** NEW 45

A stalled-screen guide now walks users through recovery, performing one automatic restart per recovery cycle.

– Explains mechanism of one restart per cycle. v0.34.0

14 **Tool call hiding in chat view** NEW 45

Tool calls can now be hidden in the chat view, matching the CLI's focus view behaviour.

– Explains behaviour and parity with CLI focus view. v0.33.1

15 **Queue editor for follow-up messages** NEW 40

Queued follow-up messages can now be merged, reordered, and steered before they are sent, via a new queue editor.

– Explains capability but no named UI path or config. v0.34.6

16 **Copy, fork, and send-time message metadata** NEW 40

Each chat message now shows copy, fork, and send time metadata beneath it.

– Names the three metadata items but no mechanism. v0.33.2

17 **Entity storage framework with auto migration** NEW 30

A new entity storage framework handles data with automatic migration.

– Bare mention of a backend framework with little detail. v0.33.3

18 **Sleep prevention when lid is closed** NEW 30

The machine is kept awake even when the lid is closed, via sleep prevention.

— Bare statement of behaviour with no configuration detail. v0.33.1

WAS THIS USEFUL? 👍 👎

palot

Feedback ⓘ v0.19.0 NOTES ↗ RANK

Just another desktop client for OpenCode 2. Manage projects, sessions, parallel agents, requests, and changes on Linux and macOS.

> Palot 0.19.0 introduces downloadable Linux and macOS desktop packages verified through GitHub build attestations, adds flexible OpenCode setup options, and drops auto-update support for legacy Palot 0.11 across this window's releases.

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

01 **Build attestation verification for downloads** NEW 60

Downloads can be verified using `gh attestation verify` against GitHub build attestations.

– Names exact command but no further verification scope given v0.19.0

THINNER COVERAGE BELOW

02 **Desktop packages for Linux and macOS** NEW 55

Adds Linux x64 desktop packages: `AppImage`, Debian/Ubuntu `DEB`, Fedora `RPM`, and portable `tar.gz`. Also adds macOS Apple Silicon desktop packages: `DMG` and `ZIP` (ad-hoc signed; not Developer ID signed or notarized).

– Names every package format but no install mechanism detail v0.19.0

03 **Flexible OpenCode setup during installation** NEW 35

OpenCode can be supplied as an existing local installation or downloaded as a verified fallback during setup.

– Describes the option but not the setup flow mechanics v0.19.0

04 **Manual updates only for legacy Palot 0.11** BREAKING 25

No auto-update feeds are published for legacy Palot 0.11 installations; updates are now manual only.

– States the change but gives no migration steps v0.19.0 v0.18.0 v0.17.0

└──▷ **BREAKING ON UPGRADE**

! No auto-update feeds are published for legacy Palot 0.11; updates are manual only.

! No legacy Palot 0.11 auto-update feeds are published; updates are now manual.

! No auto-update feeds are published for legacy Palot 0.11 installations; updates are manual.

WAS THIS USEFUL? 👍 👎

jcode

Feedback ⓘ v0.99.1 NOTES ↗ RANK

High performance coding agent harness written in Rust, built for heavy multi-agent dev workflows

> This window's jcode releases center on resilience and speed — automatic account failover, idle-session hibernation, and faster large-session resume — alongside new provider and search integrations (Venice AI, Kagi, custom HTTP JSON search), remote MCP server connectivity over HTTP/SSE, a `/telemetry` usage-collection control panel, and a string of CLI and config additions including `/reset usage limits claude`, `display.resume_project_scope`, `[menubar] enabled = false`, and Ctrl+Enter command queuing.

└──▷ **WHAT SHIPPED · 33 FEATURES** most completely described first ? what's the number?

01 **display.resume_project_scope config key** NEW 85

Adds the `display.resume_project_scope` config key so `/resume` opens scoped to the current project.

Scope the `/resume` session picker to only show sessions from the current project, reducing noise in multi-project workflows.

toml

```
[display]
resume_project_scope = true
```

— Exact config key with a runnable snippet. v0.95.0

02 **Usage telemetry controls at /telemetry** NEW 80

Adds collection controls for usage telemetry at `/telemetry`, with optional Anthropic or OpenAI OAuth account email inclusion, 90-day retention, and a conditional one-time TUI notice.

— Names the endpoint, retention window, and notice behavior. v0.97.0

03 **Menu bar helper disable config** NEW 75

Disables the macOS menu bar helper via the `[menubar] enabled = false` config setting.

Prevent the macOS menu bar helper from launching automatically, useful on shared or headless macOS machines.

```toml
[menubar]
enabled = false
```

— Exact config snippet, ready to use. v0.95.0

04 **Ctrl+Enter command queuing** NEW 75

Adds Ctrl+Enter to queue slash and ! commands to run after the current turn, with /clear dropping anything still queued.

— Exact hotkey and behavior, ready to use. v0.94.0

05 **Custom HTTP JSON search providers** NEW 70

Adds support for custom HTTP JSON search providers with configurable authentication, result mapping, and fallback engines.

— Names configurable surfaces but not the exact config keys. v0.97.0

06 **/reset usage limits claude command** NEW 70

Adds /reset usage limits claude command to reset Claude session limits from the CLI.

— Exact runnable command given. v0.96.0

07 **Alt+G git log viewer** NEW 70

Adds Alt+G shortcut to open the project's git commit log, with diff cycling moved to /diff.

— Exact hotkey and command given. v0.94.0

08 **Remote MCP server connections over HTTP/SSE** NEW 65

Supports connecting to remote MCP servers over HTTP or SSE, including OAuth sign-in and command-line management.

— Names protocols and auth but not exact CLI commands. v0.96.0

09 **Turn-end token usage hooks** NEW 65

Exposes turn-end token usage in hooks, and fires turn_start / turn_end events for one-shot jcode run turns.

— Names event hooks but no hook API signature. v0.95.0

10 **Swarm watchdog and headless socket sessions** NEW 65

Adds a stalled-worker watchdog to Swarm, and allows headless run --socket sessions to join swarms.

– Names the `--socket` flag but not watchdog thresholds. v0.95.0

11 **OpenAI Ultrafast tier for GPT-6.1 Sol** NEW 60

Supports OpenAI Ultrafast speed tiers for GPT-6.1 Sol, with Fast commands and hotkeys.

– Names model and tier but not exact command syntax. v0.99.0

12 **Markdown responses via user_response tool** NEW 60

Enables Desktop-capable clients to receive Markdown responses through the new `user_response` tool.

– Names the tool but gives no usage example. v0.98.1

13 **Expanded command risk assessment** IMPROVED 60

Expands command risk assessment to inspect commands inside $(...), backticks, and heredocs.

– Names syntax forms inspected but no severity scale. v0.95.0

THINNER COVERAGE BELOW

14 **Faster session resume and reload** IMPROVED 55

Session reload shows the local transcript immediately and sends only new history instead of reloading large sessions fully; the resume picker also opens immediately while session scanning continues in the background.

– Clear before/after but no measured speedup. v0.99.0 v0.98.1

15 **Account router and login picker enhancements** IMPROVED 55

Shows saved accounts, cached quotas, and available resets in the startup account router and usage cards, and offers additional OAuth accounts in the login picker with reauthentication of saved accounts from it.

– Describes UI elements without exact navigation path. v0.99.0

16 **Swarm mode simplification** BREAKING 55

Simplifies swarm mode by removing the task graph and defaulting swarm root effort to low.

– Names removed mechanism and new default behavior. v0.99.0

17 **SDK native runtime command invocation and session transfer** NEW 55

Rust and TypeScript SDK clients can now invoke native runtime commands and transfer sessions without converting commands into agent prompts.

— Describes capability but no API signature given. v0.97.0

18 **Saved OAuth account selection per session** NEW 55

Adds saved OAuth account selection per session from the model and login pickers, with email labels and cancelable account-switch countdowns.

— UI behavior described but no navigation path. v0.96.0

19 **OpenCode Go models routed over Responses API** IMPROVED 55

Routes OpenCode Go models (including `grok-4.7` and `gpt-6-luna`) over the Responses API.

— Names specific models and API, no setup step. v0.95.0

20 **Automatic account failover on usage limits** IMPROVED 50

Automatically retries on the next pool account when one hits a usage limit, including for Desktop and SDK clients.

— Explains behavior and client scope but no config or command. v0.99.0

21 **Idle session hibernation** NEW 50

Hibernates idle sessions to free memory, with RAM status reporting and transparent wake on the next interaction.

— Mechanism described but no config key or threshold. v0.99.0

22 **New AI provider and login integrations** NEW 45

Adds Venice AI as a supported AI provider, a configurable Kagi web search backend, and native Muse subscription login.

— Three named integrations but no setup or config detail given. v0.99.1

23 **Compressed request bodies for OpenAI and Anthropic** IMPROVED 45

Compresses OpenAI and Anthropic request bodies to reduce upload time.

— States effect but not compression method or ratio. v0.99.0

24 **SDK session collections and hidden reminders** NEW 45

Enables SDK clients to organize sessions into collections and attach hidden reminders to individual turns.

— Describes capability without API details. v0.96.0

25 **Harness API mid-turn message control** NEW 45

Harness API clients can now observe and cancel messages sent mid-turn.

— Describes capability without method names. v0.94.0

26 **Selective MCP server reconnection** IMPROVED 40

Reconnects only changed MCP servers instead of restarting all of them.

— Clear behavior change but no command or metric. v0.99.0

27 **Automatic memory recall disabled by default** BREAKING 40

Disables automatic memory recall while retaining explicit memory search.

— Clear behavior change but no migration guidance. v0.98.1

28 **Onboarding detects existing AI tool logins** NEW 40

Onboarding detects accounts signed in with Cursor and other AI tools, as well as stored Jcode agent logins.

— States detection scope but no setup steps. v0.94.0

29 **Reduced idle CPU in repos with many untracked files** IMPROVED 35

Reduces idle CPU and disk reads in repositories with many untracked files.

— States effect only, no mechanism or benchmark. v0.98.1

30 **Shared model catalog across equivalent clients** IMPROVED 35

Shares model catalog processing across equivalent clients while keeping usage and account state fresh.

— Vague on what counts as an equivalent client. v0.98.1

31 **Live per-worker stats in Swarm** NEW 30

Shows live stopwatch, context, and token stats per swarm worker.

— Bare description with no mechanism or location. v0.99.1

32 **Compact context usage display** NEW 30

Shows compact context usage before and after compaction.

— Short description with no UI location given. v0.98.1

33 **Jcode inference spending balance display** NEW 25

Shows Jcode inference spending balance.

— Bare mention with no location or mechanism. v0.99.0

WAS THIS USEFUL? 👍 👎

Replit Agent

Feedback ⓘ announced · Oct 22, 2026 BLOG ↗ RANK

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

> Replit launched Monitoring Agent, an autonomous on-call agent that investigates, fixes, and can deploy production issues without manual intervention.

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

01 Monitoring Agent for autonomous incident response NEW

76

Replit Monitoring Agent is an autonomous on-call agent that investigates production alerts, identifies root causes, validates fixes, and can deploy them without manual intervention, using access to production logs and metrics to investigate errors, slowdowns, and failed deployments in context. Enabling `auto-fix` in Publishing settings makes the agent automatically investigate and validate a fix when an alert fires and email a summary of what happened and what changed; enabling `auto-publish` in Publishing settings lets the agent deploy a validated fix to production automatically, with the creator choosing whether to allow this.

Enable fully autonomous incident response so the Monitoring Agent investigates, fixes, and ships repairs to production while you sleep.

📍 In your app's Publishing settings, enable 'monitoring alerts and auto-fix', then turn on `auto-publish` to allow Monitoring Agent to deploy validated fixes automatically.

– Names exact settings and behavior, UI path given but no API/CLI.

launch-20261022-af82b9f1

WAS THIS USEFUL? 👍 👎

pstack-claude

Feedback ⓘ v0.9.101 NOTES ↗ RANK

Claude Code, Codex, Copilot, Pi, OpenCode, Muse, Gemini, and Prime Agent versions of Poteto's pstack. Rigorous agent workflows with Cursor primitives translated for other harnesses.

> pstack-claude's biggest change this window is a breaking overhaul that routes `@level` effort settings to native per-runtime fields across Claude Code, Codex, Copilot and Pi while removing the old generated effort agents; alongside it the tool gained full Muse harness support, live background-agent monitoring via `list_agents` and `/pstack-agents`, a new `worker` model tier, a Graphite-free `orch frontier set`, and a batch of new skills including `/ketchup`, `/thermos`, `/thermo-nuclear-review`, `/sharpen` and `/in-your-own-words`.

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

01 **Live background agent monitoring** NEW 96

Adds an `activity` field to each agent's `list_agents` entry showing the current or last tool call plus a one-line argument summary and the first line of the agent's latest message, and a `sessionFile` field reporting the child's transcript path as `<timestamp>_<session id>.jsonl`. A footer line names the count of running agents and the most recently active one (e.g. `pstack: 2 running · a1b2c3d4 bash: ls -la`), visible while any agent runs. The new `/pstack-agents` command shows each agent's status, activity, transcript path, and the command to read it (`tail -f` while running, `pi --export` once ended); the session file path also appears on a `transcript:` line in the completion notice and a foreground agent's result.

— Runnable commands and exact fields given v0.9.90 v0.9.89

02 **`@level` effort routing across runtimes** BREAKING 93

Propagates `@level` on a role to the child dispatch's native effort field on every supported runtime: `effort` on Claude Code `Agent` calls, `reasoning_effort` on Codex `spawn_agent`, `effort` on Copilot `task`, and `effort` on Pi `agent` tool (passed as `--thinking`, erroring and naming valid levels if given one outside `models.json`). Removes the ten generated `effort-<level>` and `poteto-agent-<level>` agents and the `agents` list from the Claude Code plugin manifest, leaving exactly `comment-sicko`, `poteto-agent`, and the two thermo-nuclear subagents under `agents/`; `effort` frontmatter on agent files is no longer read, and the generator now fails if an `agents` list reappears in the manifest. Requires Claude Code 2.1.292 or later. `poteto-mode` also gained a rule matching each subagent's effort to the job, moving the effort level through `@level` values and effort agents in the Reasoning effort section of `plugins/pstack/skills/poteto-mode/SKILL.md`.

— Full mechanism and breaking scope across all runtimes named v0.9.95 v0.9.86

03 **Native Muse plugin build** NEW 83

Adds a Muse build with a native plugin manifest under `.muse-plugin/` listing every shared skill and a `SessionStart` hook that injects the poteto-mode mandate in Muse wording once the user approves the plugin's hooks. Adds `muse-tools.md` mapping Claude tool names, subagents, transcripts, and Muse model IDs, a `muse` block in `models.json` with a model per tier, Muse agent definitions for the four agent files (role effort suffixes passed as a Workflow child's `effort` field), and `setup-pstack` writing the Muse sheet under the Muse config directory read by the `SessionStart` hook.

— Named config paths but confined to a single release v0.9.96

04 **`worker` model tier for routine roles** NEW 83

Adds a `worker` tier to `plugins/pstack/models.json` (value `sonnet`), moves the `feature` and `refactoring` roles from the `opus` tier to `worker`, and extends `poteto-mode` to route trivial mechanical edits to the worker model while still sending the hardest changes to the strongest-judgment role. Adds a `worker` example entry (`gpt-6-sol`) to the Codex block and names worker roles explicitly in the Codex and Copilot mappings, with Copilot's setup now asking a worker-model question after the default model. Upstream defaults are now `claude-opus-5-5-xhigh` for judgment and synthesis roles and `grok-4.7-xhigh-fast` for code delegates, explorers, investigators, reflect tooling reviewer, and swarm workers, at commit `99743e9`.

— Config keys and values named, single release v0.9.88

05 **Ignored-file detection in worktree audit** IMPROVED 82

Adds `--ignored=matching` to the `git status` call in `worktree-audit.mjs` to detect ignored files (such as `.env` or `.claude/settings.local.json`) outside the `DISPOSABLE` directory list (`node_modules`, `.venv`, `__pycache__`, `dist`, `build`,

`target`), reporting them as `ignored:N` and holding the worktree as `hold-ignored` so `git worktree remove` cannot silently destroy them. The worktree-cleanup playbook now names the `hold-ignored` and `hold-reflog` labels alongside the commands that list what each count covers.

— Exact flag, labels and script named v0.9.87

---

06 **`orch frontier set` drops Graphite for `gh`/Origin CLI** BREAKING

80

`orch frontier set` now walks the PR base chain from the checkout's current branch through `gh`, or through Origin's CLI with `--forge origin`, removing the Graphite (`gt`) dependency; `gt` is removed from the dependency table in `docs/reference.md`, the Pi equivalence table, and the Codex, Pi, and Copilot mappings, so setups relying on Graphite must switch to `gh` or Origin's CLI. Each `gh pr list` call now lists up to 1000 pull requests instead of gh's default of 30.

— Exact flags and migration path named v0.9.86

---

07 **Session-based transcript directory lookup** IMPROVED 78

`claude-project-dir.mjs` now uses `CLAUDE_CODE_SESSION_ID` to find the transcript directory so skills follow the session rather than the shell's current directory; called without an argument and no `CLAUDE_CODE_SESSION_ID` set, it falls back to deriving the project name from the current directory, and called with an argument it derives the project name from that directory and skips the session lookup entirely. Trimming of `CLAUDE_CONFIG_DIR` was removed so the value is preserved exactly as Claude Code sets it.

— Exact file and env var named but single narrow fix v0.9.101

---

08 **`/thermo-nuclear-review` and `/thermos` audit skills** NEW

76

`/thermo-nuclear-review` audits a branch's changes for bugs, breakage, security issues, devex regressions, and feature-gate leaks. `/thermos` runs the thermo-nuclear audit and a code-quality review as two background subagents (`pstack:thermo-nuclear-review-subagent` and `pstack:thermo-nuclear-code-quality-review-subagent`) over the same diff and file contents, then synthesizes their findings. Syncs thermos 1.0.0 as its own component at commit `118e73b`, bringing `thermo-nuclear-code-quality-review` from thermos instead of cursor-team-kit.

— Named subagent ids and commit, no UI path given v0.9.83

---

09 **New guidance slash commands** NEW 65

Adds `/ketchup`, a port of upstream Cursor's `/visualize` command, which catches the user up on everything in the chat since their last message in about 150 words built on real visuals, with a before-and-after visual for each action item, autopilot ticks and

replies, and charts rendered from data; `/in-your-own-words`, which plays back the user's goal and the problem before work starts; and `/sharpen`, which frames a task the way a top expert would without overspecifying it. `/poteto-help` now routes to all three.

— Behaviour and lineage described, no config surface Commands and routing named but thin on mechanism

v0.9.86 v0.9.83

10 **`watch-pr` review-gate and pagination fixes** IMPROVED

64

`watch-pr` now reads a Code Review Gate check that concludes `ACTION_REQUIRED` as the owner-approval gate, and reads review threads one page at a time via a cursor-walking `resolveReviewThreads` loop, preserving full Bugbot pass counts across paginated threads.

— Named mechanism and check but no direct usage path v0.9.86 v0.9.83

THINNER COVERAGE BELOW

11 **Skill resolution safeguards against stale installs** IMPROVED

59

When a delegate agent must find a skill itself, it now picks the newest install by `.claude-plugin/plugin.json`; a substitution rule and a denylist token exclude upstream's `.cursor-plugin/plugin.json`. The sync denylist also now rejects `/visualize`.

— Named manifests and denylist, mechanism sketched briefly v0.9.86

12 **Stricter principle checks for test behavior and proof of work**

IMPROVED

58

Updates `principle-test-behavior-not-implementation` to evaluate whether a test would pass if every imported function returned `undefined`, names five shapes that would, and adds an absence-only test and a constant pin to the keep-list; `poteto-mode` summaries of `principle-attack-the-premise` and `principle-test-behavior-not-implementation` were updated to follow upstream text. Extends `principle-prove-it-works` with a check requiring the model to read real state and the exit code before reporting success, rather than relying on truncated output or its own echo.

— Names the principle and check but no usage command States the rule but no further surfaces

v0.9.82 v0.9.88

13 **Root `plugin.json` Agent Plugins 1.0.0 manifest** NEW

57

Ships a root `plugin.json` at the package root conforming to the Agent Plugins 1.0.0 manifest format, with the port's author, homepage, and repository fields, version-

stamped by the generator.

– Names format and fields but no usage instructions v0.9.83

14 **Namespaced subagent IDs and CI registration check** IMPROVED 57

Adopts `pstack:comment-sicko` as the namespaced subagent id (replacing upstream's `subagent_type: 'comment-sicko'`) so Claude Code can dispatch the agent, and adds a new CI test verifying every agent under `agents/` is named in the Pi agent tool description and in the Pi, Copilot, and Codex subagent policies.

– Internal ids and CI check named, low reader action v0.9.83

15 **Fence tracking fix in `check-plan.mjs`** IMPROVED 43

Updates `check-plan.mjs` with fence tracking by delimiter character and length, keeping opening fences at any indentation so fences inside list items are counted correctly.

– Names the file and fix but narrow in scope v0.9.83

16 **Pi `haiku` role updated to `claude-haiku-5-5`** IMPROVED 35

Updates the `anthropic` provider's `haiku` role to `claude-haiku-5-5` on Pi, matching the opus, fable, and sonnet generations.

– Exact model id named but minimal context v0.9.82

17 **`principle-fix-root-causes` principle** NEW 30

Adds `principle-fix-root-causes`, which proposes a cheap always-on signal after a deep dig.

– Bare description with no mechanism detail v0.9.86

└──▷ BREAKING ON UPGRADE

! The ten generated agents `effort-<level>` and `poteto-agent-<level>` are removed; any workflow referencing them by name will break.

! The `agents` list is removed from the Claude Code plugin manifest; Claude Code now registers the default `agents/` directory instead.

! `effort` frontmatter in agent files under `agents/` is no longer read; any file that set it will have that value silently ignored.

! `@level` requires Claude Code 2.1.292 or later; older Claude Code versions are not supported for this feature.

! Orchestrate no longer uses Graphite: `gt` is removed from the dependency table in `docs/reference.md`, the Pi equivalence table, and the Codex, Pi, and Copilot mappings; setups relying on Graphite for `orch frontier set` must switch to `gh` or Origin's CLI with `--forge origin`.

WAS THIS USEFUL? 👍 👎

clay

Feedback ⓘ v5.3.0 NOTES ↗ RANK

A self-hosted workspace where people and coding agents work together across projects, sessions, and providers.

> Clay's three releases this window centered on a major workspace overhaul — persistent split-view multi-session panes, passkey/password authentication, a hardened Knowledge store, consent-based MCP vendor access, sandboxed HTML previews, and a rebuilt settings UI — alongside a breaking change to OS-user group inheritance that retires the legacy `inheritGroups=false` setter.

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

01 Mates personalization and scheduling NEW — 70

Adds approved scheduling for Mates during and after creation, Clay-curated personal memory and workspace colors, integrated personal workspaces and Clay memory, editable identity prompts in Mate settings, an inset workspace canvas extended to all Mates, and integration of Mates into project navigation.

— One-line addition, no mechanism or UI path. Enumerates many concrete additions with a clear settings location.

v5.2.0 v5.1.0

02 OS-user group inheritance change BREAKING — 70

OS-user processes now inherit account groups whenever `setpriv` is available; legacy `inheritGroups=false` values are ignored and the group-inheritance setter is retired.

— Precise before/after with exact config key and migration path. v5.1.0

03 Knowledge module enhancements NEW — 60

Adds guarded writes, resumable database imports, portable storage, guarded migration tools, and owner-controlled memory deletion to the Knowledge store, along with the Mate document workbench, `NeDB` database items, fullscreen editing, and linked documents in the knowledge module.

— Enumerates several concrete capabilities but no commands or config keys. Names NeDB and workbench features but no usage steps.

v5.2.0 v5.1.0

04 **Settings UI overhaul** IMPROVED 60

Rebuilds server administration and save workflows in settings, rebuilds project settings as a focused modal, redesigns personal settings and provider credentials UI, and unifies personal integrations and named connections in settings.

— Names several settings surfaces reorganized but no exact menu paths. v5.1.0

THINNER COVERAGE BELOW

05 **Workspace UI improvements** NEW 50

Adds persistent multi-session split-view with docked panes retained beneath tool overlays, plus view supervision and compact pane controls, along with a staged project-start dialog, user-controlled primary work links to sessions, and a date heading visibility toggle in the sidebar.

— Describes layout mechanics but no exact UI path or command. Groups three thin UI additions each named but undetailed.

v5.3.0 v5.2.0

06 **Passkey and password authentication** NEW 50

Adds passwords, passkeys, and session security to account authentication, along with remembered accounts and a refined sign-in flow.

— Lists auth methods added but no setup steps or flags. v5.2.0

07 **AGENTS.md identity storage** NEW 50

Migrates mate identity documents to `AGENTS.md`, establishing it as the identity storage format referenced across releases.

— Names exact file but no detail on migration mechanics. v5.2.0 v5.1.0

08 **Disk usage reporting** IMPROVED 50

Adds disk usage reporting, now backed by streamed native filesystem metadata for faster usage scans.

— Explains backing mechanism but no command or metric shown. v5.1.0

09 **MCP vendor access unification** IMPROVED 45

Unifies MCP vendor access and adds consent-based imports, plus vendor-specific account usage readers.

— Describes unification and consent model without endpoint or flag detail. v5.2.0

10 **Sandboxed HTML file previews** NEW 45

Adds sandboxed HTML previews to the file viewer with account-aware file access.

— Mentions sandboxing and account-aware access but no config or flag. v5.1.0

11 **Toolbar activity badges** NEW — 40

Adds sticky note and browser tab count badges to the toolbar for at-a-glance activity tracking.

— Names exact badge types and location but no deeper mechanism. v5.3.0

12 **Project debate workbench replaces home board** IMPROVED — 35

Replaces the home board with a project debate workbench.

— Notes replacement but no detail on new workbench's capabilities. v5.1.0

13 **Search action prioritization and popup** NEW — 30

Adds Clay action prioritization and conversation popup opening directly from search results.

— Thin description, no mechanism or exact UI location. v5.3.0

14 **Driver update queueing** NEW — 25

Queues reviewed updates for active Drivers in the workspace.

— Single vague sentence, no mechanism or scope. v5.3.0

15 **Browser session header toggle** NEW — 25

Adds a browser session header toggle.

— Single-line toggle addition without further detail. v5.1.0

└──▷ **BREAKING ON UPGRADE**

! OS-user processes now inherit account groups whenever `setpriv` is available; legacy `inheritGroups=false` values are ignored and the group-inheritance setter is retired.

WAS THIS USEFUL? 👍 👎

claude-code-memory-setup

Feedback ⓘ v1.1.0 NOTES ↗

RANK

5x fewer tokens per session on Claude Code with Obsidian + Graphify. Persistent memory, codebase knowledge graphs, and chat import pipeline.

> v1.1.0 restructures the session command toolkit, adding a `commands/` installation folder and `VAULT_DIR` configuration, overhauling `/save`'s consolidation and session-logging mechanism, clarifying the `/memorize` vs `/save` split, and making `/commit` auto-trigger `/save`.

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

01 Command files and VAULT_DIR configuration NEW

83

Adds a `commands/` folder to the repository as the canonical home for all four session commands (`/save`, `/resume`, `/memorize`, `/commit`), with `skills/` remaining as an alternative installation path. Sets the `VAULT_DIR` environment variable (defaulting to `~/vault`) to configure the vault path used by the session commands.

Install all four session commands into Claude Code in one step so `/save`, `/resume`, `/memorize`, and `/commit` are available immediately.

```
$ cp commands/*.md ~/.claude/commands/ && rm
~/.claude/commands/README.md
```

Point the session commands at a non-default vault location before restarting Claude Code.

```
$ export VAULT_DIR=~/Documents/my-vault
```

— Names folder, env var, default, and gives install commands v1.1.0

02 /save two-phase consolidation and session logging IMPROVED

83

`/save` now runs a two-phase consolidation: explicit criteria filter what is worth keeping, then a grep-based deduplication step runs before every write, routing each knowledge kind to its own destination and writing one-line pointers into `## Permanent facts` of `<vault>/<project>/CLAUDE.md`. Session logs written by `/save` also gain a structured State for resume section covering task, branch, pending criteria, tests, blockers, and next action.

— Mechanism and file path given, but no runnable command example v1.1.0

THINNER COVERAGE BELOW

03 **/memorize scoped to mid-session, /save owns end-of-session**

IMPROVED

50

`/memorize` is now an explicit mid-session trigger for capturing knowledge, while end-of-session consolidation is handled exclusively by `/save`.

— Explains division of responsibility but no mechanism detail v1.1.0

04 **/commit auto-runs /save after push** IMPROVED

50

`/commit` now automatically runs `/save` after the push completes.

— Names both commands but no further mechanism given v1.1.0

WAS THIS USEFUL? 👍 👎

ai-git-bot

Feedback ⓘ 1.28.0 NOTES ↗ RANK

The self-hosted AI workflow automation platform for Git repositories.

> ai-git-bot 1.28.0 adds an OpenRouter AI adapter with routing and privacy controls, a new `code-execute` tool for agent-run code, and optional integer parameter ranges in workflow definitions, while removing a deprecated admin config field.

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

01 **code-execute tool for agent tasks** NEW 50

Adds a `code-execute` tool enabling the agent to run code as part of review and workflow tasks.

– Names the exact tool but not its invocation or scope. 1.28.0

02 **Removal of Bot.botType admin field** BREAKING 50

The deprecated `Bot.botType` field has been removed from the admin configuration.

– Names exact field removed but no migration guidance given. 1.28.0

03 **OpenRouter AI adapter** NEW 45

Adds OpenRouter as an isolated AI adapter, with configurable routing and privacy settings for how requests are routed and handled.

– Names the adapter and settings but no concrete config keys or values. 1.28.0

04 **Optional integer parameter ranges in workflows** NEW 30

Adds support for optional integer parameter ranges in workflow definitions.

– Bare description with no syntax or example given. 1.28.0

└──▷ **BREAKING ON UPGRADE**

! The deprecated `Bot.botType` field has been removed from the admin configuration.

WAS THIS USEFUL? 👍 👎

cezar

Feedback ⓘ v0.16.0 NOTES ↗ RANK

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 shipped a visual workflow node editor with a full graph engine, encrypted secrets and per-project MCP server config, plus a redesigned task workspace with Terminal, Design Mode and Preview views, a ⌘P command palette, and deeper git integration across two releases.

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

01 GitHub comment body no longer required in list responses

90

BREAKING

GitHub list rows no longer require `body`; it is optional during a deprecation window and will come from the detail payload — consumers should hydrate full text from `GET /api/v1/github/comments/:kind/:number` (`detail.body`).

— Exact endpoint, field, and migration path given v0.15.0

02 Visual workflow node editor and graph engine NEW

85

Introduces a visual workflow node editor and graph engine supporting loops, gates, dispatch/fork/join, composition, git and GitHub nodes, YAML import/export, simulation, and a live graph on task runs; existing v1 YAML workflows continue to run unchanged.

— Mechanism and scope well described, but UI-driven not a direct command v0.15.0

03 Private per-project MCP servers NEW

75

Private per-project MCP servers can be configured in `.ai/cezar/mcp.local.json`, which is git-ignored and passed to every agent that accepts MCP at launch.

— Exact config file path given, scope and behavior clear v0.15.0

04 Encrypted secrets management NEW

75

Adds encrypted project and workspace secrets for check steps, managed via the `cezar secrets` CLI and Settings → Secrets; secrets are never forwarded to agents.

— Named CLI and UI path, clear guarantee stated v0.15.0

05 PR-head checkout for automations NEW

75

Automations can set `checkout: 'pr-head'` to verify a pull request's head commit rather than the base branch.

Trigger an automation that checks out a PR's actual head commit rather than the base branch, so the agent reviews what will actually be merged.

```yaml
checkout: 'pr-head'
```

— Exact config key with a runnable usage example v0.15.0

06 **CLI and API improvements** NEW 70

Adds `cez runs list` to list runs from the terminal, a human-readable error when `CEZ_API_URL` is missing, and published `/api/v1` API docs.

— Named command and env var are directly runnable v0.15.0

07 **New workspace views: Terminal, Design Mode, Preview** NEW 70

The task workspace gains Terminal as a first-class view with a task-tab row and named layouts saved per task, a Design Mode, and a Preview view that renders as a Browser so agents can serve the running app directly inside the cockpit; per-project start recipes let an empty Preview launch the task's app without manual setup.

— Mechanism and scope clear but entry point is UI navigation only v0.16.0

08 **Git integration and Git tab improvements** NEW 65

Git tab gains Source-Control sync for the main checkout plus in-place task creation on a new branch via a new git integration, agent-driven git steps with live loading states shown in the run header, and a repository file browser (file tree, viewer, and path filter).

— Several named UI surfaces but no commands or configs v0.16.0 v0.15.0

09 **Attachment staging on compose** NEW 65

Compose now supports staging up to 20 files per message and 20 MB of images, accepting HTML, CSV, JSON, and source files alongside images and PDFs.

— Concrete limits and formats named, UI-only action v0.16.0

10 **Changes tab improvements** IMPROVED 60

The Changes tab is now instant to revisit and responsive on large diffs; it also adds diff-line commenting that sends collected comments to the agent as a single review, plus a Markdown preview toggle in the diff view.

— Behavior described but accessed only via UI v0.16.0 v0.15.0

11 **⌘P task palette** NEW 60

A `⌘P` task palette in the web UI lets you act on tasks, fast-switch between them, and use keyboard shortcuts.

— Exact shortcut named, immediately usable v0.16.0

THINNER COVERAGE BELOW

12 **Task list sorting and filtering** IMPROVED 55

Tasks list adds sorting by last activity with a Recent/Created toggle, and a task origin filter (Regular, Automations, All) defaulting to Regular to separate automation-started tasks from manually started ones.

— Named toggles and defaults, UI-only entry point v0.16.0 v0.15.0

13 **Settings UI overhaul and cockpit branding** IMPROVED 55

Settings move to a full-screen surface with a shared sidebar, replacing previous per-section settings screens; Appearance settings now let you customize cockpit branding with a display name and logo.

— Clear navigation path, limited mechanism detail v0.16.0 v0.15.0

14 **Start/Plan switch removed from composer** BREAKING 50

The composer's Start | Plan first switch and the plan review overlay are removed; Start now always creates a run immediately.

— Clear before/after but no migration mechanism given v0.15.0

15 **Idle-time self-update** NEW 40

cezar now applies updates automatically during idle time, when no tasks are running across any project.

— Behavior described but no configuration surface given v0.16.0

16 **Lifecycle script recovery in worktrees** IMPROVED 40

Lifecycle scripts in worktrees now get explicit recovery: failures are surfaced and retried on error.

— Behavior stated without scope or configuration detail v0.16.0

17 **Unified Skills panel** NEW 35

Adds a unified panel for browsing, activating, and previewing Skills.

— Thin description with no mechanism or named surface v0.15.0

└──▷ BREAKING ON UPGRADE

! The composer's Start | Plan first switch and the plan review overlay are removed; Start always creates a run immediately.

! GitHub list rows no longer require `body`; it is optional during the deprecation window and will come from the detail payload — consumers should hydrate full text from `/api/v1/github/comments/:kind/:number` (`detail.body`).

WAS THIS USEFUL? 👍 👎

Alibaba OpenCodeReview

Feedback ⓘ v1.13.2 NOTES ↗

RANK

Hybrid code review tool combining deterministic pipelines with LLM agents for precise line-level comments. Built-in multi-language rulesets cover NPE, XSS, SQL injection, and more.

> OpenCodeReview v1.13.0 adds support for ChatGPT plan reviews through the public Responses API and introduces ACP (Agent Communication Protocol) service integration.

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

01 **ChatGPT plan reviews via Responses API** NEW 35

OpenCodeReview now enables ChatGPT plan reviews through the public Responses API.

— Names the API but no setup detail or config shown v1.13.0

02 **ACP service integration** NEW 25

Adds ACP (Agent Communication Protocol) service support for OpenCodeReview.

— Names the protocol but no mechanism or usage given v1.13.0

NEW SUPPORTED PLATFORM Shader Languages support

NEW SUPPORTED PLATFORM Nushell language support

NEW SUPPORTED PLATFORM PowerShell review support

v1.13.2 NOTES ↗ LATEST STABLE

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v1.13.2 https://github.com/alibaba/open-code-review.git
# already have the repo? check out this version:
$ git checkout v1.13.2
```

WAS THIS USEFUL? 👍 👎

zentty

Feedback ⓘ v0.5.0 NOTES ↗ RANK

A native macOS terminal for agent-driven development, built on Ghostty.

> Zentty's biggest update this window adds a full CLI for controlling agent panes — forking, reading, sending input, and streaming live events — alongside Muse Code agent integration, persistent terminal overlays, and smaller additions like unread tracking for idle panes and CLI-driven folder opening.

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

01 **Sidebar and pane status indicators** NEW 79

Adds sidebar progress percentages, paused states, and errors for programs that report terminal progress, plus unread tracking via ⌘U to mark an idle agent pane as unread with a magenta Check back status and ⇧⌘U to jump back to it, with panes also markable as read from the sidebar, command palette, or Navigation menu.

Flag an idle agent pane for follow-up so you don't lose track of it across multiple active sessions.

> 📍 Press ⌘U on an idle agent pane to set its status to Check back (shown in magenta). Later, press ⇧⌘U to jump directly back to it, or open the sidebar, command palette, or Navigation menu to mark it read.

— No named surface or steps to enable it. Exact shortcuts and alternate UI paths given with example.

v0.5.0 v0.3.6

02 **zentty pane commands** NEW 78

Adds `zentty pane fork` to fork an agent conversation into a new pane or worklane (supporting Claude Code, Codex, OpenCode, Droid, Grok, Pi, and Kilo) with configurable placement, plus `zentty pane send` and `zentty pane send-key` to send text and named keys to panes (text submission waits for the agent to accept input unless forced), and `zentty pane read` to read a pane's visible screen or scrollback as text or JSON.

— Command and agent list named but no example invocation shown. Three commands named with behaviour but no example invocation.

v0.5.0

03 **Agent launch flags for teams and tmux** NEW 76

`zentty launch --teams` adds team support and `zentty launch --tmux-shim` adds tmux compatibility when launching agents, and the two flags can be combined for orchestrator-driven workflows.

Launch an agent with team support and tmux compatibility in a single command for orchestrator-driven workflows.

```
$ zentty launch --teams --tmux-shim <agent>
```

— Exact flags and runnable example command provided. v0.5.0

04 **Config storage and cross-Mac settings sync** IMPROVED 76

Local session and restore state moved from `~/.config/zentty` into `~/Library/Application Support/zentty` so the config directory is safer to sync between Macs; background opacity, custom notification sounds, and home-relative app and browser paths now sync across Macs through synced settings.

Verify that local session state has moved out of your synced config folder after upgrading, so it no longer causes conflicts across Macs.

> 📍 Steps: Open Finder and confirm that per-machine session and restore state now lives in ~/Library/Application Support/zentty rather than ~/.config/zentty.

— Exact paths named with a verification step included. v0.4.0

05 **Folder opening and split-pane startup via CLI** NEW 71

Open a folder in a new terminal window via `open -a Zentty <dir>`, by dropping it on the Dock icon, or through Finder's Open With menu; startup commands can also be run in split panes via the CLI.

Open a project directory directly in Zentty from the shell without navigating through the UI.

```
$ open -a Zentty <dir>
```

— Runnable command given; split-pane CLI startup lacks its own example. v0.3.5

06 **Live event streaming via zentty events** NEW 70

`zentty events` streams worklane, pane, agent status, and attention changes in real time, for driving external automation or alerting.

Watch all worklane and agent status changes in real time to drive external automation or alerting.

```
$ zentty events
```

— Runnable command shown via usage example. v0.5.0

THINNER COVERAGE BELOW

07 **Expanded tmux compatibility for agent teams** IMPROVED

59

Expanded tmux compatibility for agent teams and tmux-based orchestrators adds session and window commands, pane targeting, format expressions, shell commands, and pane respawning.

— Lists covered tmux features but no exact commands given. v0.5.0

08 **Muse Code agent integration** NEW

50

Muse Code integration covers agent status, tasks, questions, approvals, subagent tracking, and session restore.

— Lists covered aspects but no setup steps given. v0.4.0

09 **Persistent terminal overlays** NEW

47

Persistent terminal overlays can be scoped to a pane or worklane, with sessions that keep running when hidden or when switching worklanes.

— Describes behaviour but no activation steps named. v0.4.0

10 **Reply to agents from notification banners** NEW

42

Reply to finished agents directly from notification banners, a behaviour that can be toggled in Settings > Notifications.

— UI path given but no further mechanism detail. v0.5.0

WAS THIS USEFUL?  

flow-next

Feedback ⓘ flow-next-v9.1.0 NOTES ↗ RANK

Workflow plugin that transforms bugs, ideas, or tickets into verified pull requests with specs, cross-model review, live QA, and receipts. Works with Claude Code and more.

> flow-next's biggest addition this window is Goal mode, which autonomously drives multi-spec outcomes from a PRD, spec, or plain-language goal through to a merged pull request; alongside it came opt-in regulated-environment support (committed evidence trails, PR-gated spec approval, a traceability matrix command), four new natural-language request routes (profile diagnosis, repeated-mistake capture, prompt evaluation, diff safety proof), native Muse Code support, and a custom flowctl binary override.

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

01 **Goal mode for multi-spec autonomous runs** NEW 96

`/flow-next:flow --goal <spec-id or source>` pursues outcomes larger than one spec — it accepts a goal spec, PRD, design document, migration target, or plain sentence, measures criteria, mints child specs, builds each through the normal pipeline, and opens a final pull request only when every criterion has recorded evidence. The `--until=merge` flag lands each child spec and the final pull request automatically under the run's merge consent; child specs land on a `goal/<branch>` integration branch kept current by merging the default branch before each measurement, and the run ends `GOAL_MET` only once every criterion has evidence. A `## Goal log` section appended to the goal spec records measurements, mints, decisions, and open questions so a multi-day run resumes from disk.

Autonomously drive a multi-day migration from a PRD to a merged PR — flow measures criteria, writes child specs, and keeps going until every criterion has evidence.

```
$ /flow-next:flow --goal migration-prd.md --until=merge
```

Run Goal mode from a single sentence when you do not yet have a formal spec — flow captures the goal, then iterates until done.

```
$ /flow-next:flow --goal 'Migrate all database queries from raw SQL to the ORM layer'
```

— Full mechanism, flags, branch convention, and log all named. flow-next-v9.0.0

02 **Traceability matrix command** NEW 95

`flowctl trace <spec-id>` prints a per-criterion traceability matrix in CSV, Markdown, or JSON (`--json`), maps test results from a JUnit report by R-ID, flags every criterion without evidence and every piece of evidence that names no criterion, and exits `1` on any gap.

After a feature is complete, generate a full traceability matrix in Markdown to verify every acceptance criterion has test coverage before submitting for audit.

```
$ flowctl trace SPEC-42
```

Export the traceability matrix as machine-readable JSON for ingestion into an audit or compliance reporting pipeline.

```
$ flowctl trace SPEC-42 --json
```

— Exact command, flag, formats, and exit code. `flow-next-v8.7.0`

---

03 **Committed evidence trails via evidence.persist** NEW

93

Setting `evidence.persist on` (with optional custom `evidence.dir`) makes `flowctl` write and git-stage a copy of every plan, implementation, completion review receipt, QA receipt with captures, and test-runner report into `.flow/evidence/<spec-id>/` at the moment the record is written; a later round never overwrites an earlier one, and a failed write fails the stage.

In a regulated project, enable committed evidence trails and PR-gated spec approval so every artefact is git-staged at write time and no spec can be implemented without a reviewed merge.

yaml

```yaml
evidence:
  persist: on
  dir: .flow/evidence

spec:
  approval: pr
```

— Exact config key, path, and write semantics. `flow-next-v8.7.0`

---

04 **PR-gated spec approval via spec.approval** NEW

93

Setting `spec.approval pr` gates implementation on the spec pull request (branch `spec/<spec-id>`) receiving at least one approving review before merge; `spec ready` ,

`start`, and `flow --auto` all refuse an unapproved spec, and once the PR merges the next selection records the approvers, merge identity, and timestamps automatically.

In a regulated project, enable committed evidence trails and PR-gated spec approval so every artefact is git-staged at write time and no spec can be implemented without a reviewed merge.

```yaml
evidence:
  persist: on
  dir: .flow/evidence

spec:
  approval: pr
```

— Exact config key, branch pattern, and gating behavior. `flow-next-v8.7.0`

---

05 **Custom flowctl binary routing** NEW 91

Setting `FLOW_NEXT_FLOWCTL` redirects every flowctl call — across all skills, agents, and hooks — to a custom, forked, or alternative-language flowctl binary. Both `scripts/flowctl` (bash/Git Bash/WSL) and `scripts/flowctl.cmd` (cmd/PowerShell) launchers forward all arguments and return its exit code; if the named binary cannot be run, the launcher halts with `flowctl: FLOW_NEXT_FLOWCTL is set but could not be run` instead of silently falling back to the bundled flowctl.

Test a forked or locally built flowctl binary across all skills and agents without modifying any skill files or copying anything into your repository.

```
$ FLOW_NEXT_FLOWCTL=/path/to/my-fork/flowctl /flow-next:flow
'Add passwordless login'
```

Persist a custom flowctl for every session in your shell profile, so every flow-next invocation uses your build automatically.

```
$ export FLOW_NEXT_FLOWCTL=/usr/local/bin/flowctl-custom
```

— Names exact env var, launcher scripts, and failure message. `flow-next-v9.1.0`

---

06 **Native Muse Code support** NEW 88

`/flow-next:setup` now detects a Muse Code install, writes instructions to `AGENTS.md`, and recommends a CLI review backend; setup is via `./scripts/install-muse.sh` followed by launching with `muse --disable-sandbox`, giving access to all 21 agents. `muse skills user-only plan --scope built-in` stops Muse's own

planner from intercepting natural-language `plan X` requests so flow-next handles them instead.

Set up flow-next in Muse Code so all 21 agents are available — run the installer, launch Muse with sandbox disabled, then initialise the project.

```
$ ./scripts/install-muse.sh
muse --disable-sandbox
# In your project's Muse conversation:
/flow-next:setup
```

Stop Muse's own planner from intercepting natural-language `plan X` requests so flow-next handles them instead.

```
$ muse skills user-only plan --scope built-in
```

— Exact install commands and flags given. `flow-next-v8.6.0`

---

07 **New natural-language request routes for flow** NEW 79

Flow gains new natural-language request routes: `/flow-next:flow <path-to-heap-snapshot-or-crash-dump>` diagnoses a heap snapshot or crash dump without touching code, for agents hitting an OOM or an unexplained crash in CI; telling flow that agents keep making the same mistake captures a spec of checks that fail on a past occurrence, following a hardening ladder from removing the opening through types and a lint rule down to instruction text only for judgment calls; asking flow whether a prompt or skill change behaves better returns a measured verdict — promote, do not promote, or inconclusive — using your existing eval harness or a small blinded one it builds with a judge from another model family; asking flow what a diff could break returns the one fact it is safe because of, proven by running code, with a proof depth (stated, pointed at, walked, ran, reproduced), each risk paired with the check that would catch it, and what was checked and cleared; and refactors of UI meant to look the same are pinned by screenshots, with the refactor route capturing named states before the change, comparing by pixel diff, and never regenerating the baseline to pass.

Diagnose a heap snapshot or crash dump without touching code — useful when an agent keeps hitting an OOM or unexplained crash in CI.

```
$ /flow-next:flow <path-to-heap-snapshot-or-crash-dump>
```

Capture a repeated-mistake check when code review keeps catching the same pattern — flow collects evidence from commits and reverts and builds a failing-spec entry.

```
$ /flow-next:flow 'agents keep doing X'
```

Measure whether a prompt or skill change is worth promoting before rolling it out — flow runs or builds a blinded eval and returns promote, do not promote, or inconclusive.

```
$ /flow-next:flow 'does this prompt change behave better'
```

— Thin: one command, no mechanism described. Mechanism (hardening ladder) and command given. Verdict mechanism and command described. Proof-depth mechanism named but no command. Mechanism named but no command or config.

`flow-next-v8.8.0`

---

08 **Task-scoped review focus and Codex reviewer parity** IMPROVED

77

`impl-review --focus` and `impl-review-fanout --focus` now work on task-scoped reviews, so every reviewer — including fan-out reviewers — sees the declared focus, and re-reviews retain it; the review focus declared in project instructions (e.g. 'reviews focus on over-engineering and YAGNI') is now also passed to Codex reviewers, which previously ran without access to the repository's `AGENTS.md`.

Scope a review focus to a single task's fan-out review so all parallel reviewers check for the same concern without changing project-wide instructions.

```
$ impl-review-fanout --focus "over-engineering and YAGNI"
```

— Exact flags and before/after for Codex reviewers. `flow-next-v8.5.0`

---

09 **Incremental QA re-checks** IMPROVED

70

QA re-checks now only scenarios that could have changed since the last receipt, carrying passing results forward from the prior head and running a full pass only when shared code changes broadly, the acceptance contract changes, or the smoke run fails; `flowctl qa receipt` reflects carried-vs-live status per scenario.

— Names flowctl qa receipt but no exact trigger values. `flow-next-v9.0.0`

---

10 **Audit Harden picks gate from hardening ladder** BREAKING

67

Audit's Harden path now picks its gate from the hardening ladder and proves the gate fails on a past occurrence when the entry names one, instead of always writing a `CLAUDE.md` or `AGENTS.md` instruction-file rule; when the right fix changes code or the

repo has no lint or CI surface for the gate, Harden instead recommends flow's repeated-mistake route and leaves the entry active.

— Named files and before/after behavior described. flow-next-v8.8.0

11 **Single-pass base catch-up before merge in Land** BREAKING

64

Land now catches up a behind pull request with the base only once, as the last step before an authorized merge — even on repositories that do not require up-to-date branches, and only once all other gates pass — pins the update to the head it just checked, and counts patience from the author's last push, eliminating a fresh CI run on every tick.

— Mechanism and before/after described, breaking change. flow-next-v8.8.0

THINNER COVERAGE BELOW

12 **Custom base branch for spec work** NEW

55

`/flow-next:work --base=<branch>` starts a spec branch from a branch other than the default, useful when building on top of in-progress work.

Start a child spec branch from a feature branch rather than the default branch, useful when building on top of in-progress work.

```
$ /flow-next:work --base=feature/auth-refactor
```

— Single flag with example, no deeper mechanism given. flow-next-v9.0.0

13 **Reasoning-effort tiering for workers** IMPROVED

55

Workers on clearly routine tasks now run one reasoning-effort level below the host default, and on clearly hard tasks one level above; the effort used is visible in the `Tier:` line of each task's done summary.

— Explains behavior but no configurable surface named. flow-next-v9.0.0

14 **Rigorous performance measurement verdicts** IMPROVED

53

Performance claims now require each side to run at least five times, alternated, and the result is labelled `faster`, `slower`, `no measurable difference`, or `inconclusive` along with run count, range, and the named limiter.

— Describes rigor and labels but no user-facing command. flow-next-v9.0.0

└──▷ BREAKING ON UPGRADE

! Audit's Harden path no longer writes a `CLAUDE.md` or `AGENTS.md` rule for a lesson a check could enforce — when the right fix changes code or the repo has no lint or CI surface for the

gate, Harden recommends flow's repeated-mistake route and leaves the entry active instead.

! Land now brings a behind pull request up to date before merging even on repositories that do not require up-to-date branches, and does so only once all other gates pass.

WAS THIS USEFUL? 👍 👎

opencode-telegram-bot

Feedback ⓘ v0.26.5 NOTES ↗ RANK

OpenCode mobile client via Telegram: run and monitor AI coding tasks from your phone while everything runs locally on your machine. OpenCode V2 support.

> OpenCode Telegram Bot added prebuilt multi-arch Docker images with a container health-check endpoint, alongside chat display refinements covering subagent task labeling, execute-call truncation, and link-preview suppression on bot tool lines.

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

01 Prebuilt Docker images and health-check endpoint NEW

90

Prebuilt multi-arch (`linux/amd64` , `linux/arm64`) Docker images are now published to `ghcr.io/grinev/opencode-telegram-bot` for stable releases, eliminating the need to build locally. A Docker health-check endpoint marks the container `unhealthy` when the bot stops responding and reports whether OpenCode is reachable.

– Names registry, architectures, and health-check behavior precisely. v0.26.4

02 Link previews suppressed on bot tool lines IMPROVED

65

Link previews are now suppressed on bot-generated tool lines such as `🌐 webfetch <url>` , cards, prompts, dashboard, and command replies, while previews are preserved on the agent's own replies.

– Lists exact affected line types but offers no user control. v0.26.5

THINNER COVERAGE BELOW

03 Execute call truncation in OpenCode V2 Code Mode IMPROVED

55

Long `execute` call code in OpenCode V2 Code Mode is now truncated the same way long `bash` commands are truncated, preventing large scripts from flooding the chat.

– Behavior described but mechanism of truncation not detailed. v0.26.5

04 Subagent task names shown in activity line NEW

55

When a subagent launches a task, the bot now shows which task it launched in its activity line, e.g. `🤖 Running Task: Count Kotlin files` , in both normal and compact mode.

– Concrete example given but no configuration surface named. v0.26.5

WAS THIS USEFUL? 👍 👎

pairlet

Feedback ⓘ v2.6.0 NOTES ↗

Pairlet (formerly CC Pocket / cc-pocket) — Continue your local AI coding tasks from phone, tablet, or desktop.

Adds optional voice proofread-then-send on mobile with edit preview; improves session list filtering and Claude project attribution.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.6.0 https://github.com/heypandax/pairlet.git
# already have the repo? check out this version:
$ git checkout v2.6.0
```

Adds a mobile after voice input setting that enables proofread-then-send, previewing suggested edits before dispatch and flagging uncertain changes for manual review.

Filters Codex internal sub-task entries from the session list and improves Claude project session attribution after a project is moved.

NEW INTEGRATION DeepSeek Harness 0.2.0-rc.2 v4 session history support

UPDATED SUPPORTED PLATFORM macOS, Windows, Linux, Android build; iOS via App Store; HarmonyOS excluded

WAS THIS USEFUL? 👍 👎

socraticode

Feedback ⓘ v1.16.1 NOTES ↗ RANK

Enterprise-grade codebase intelligence with hybrid semantic search, dependency graphs, symbol-level impact analysis, and cross-project search. 61% fewer tokens, 37x faster.

> SocratiCode shipped result limits for its graph query tools and an opt-in Git-triggered index refresh.

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

01 **Result limits on graph queries** IMPROVED 55

Adds a limit parameter to `codebase_symbol`, `codebase_impact`, and `codebase_flow` graph queries to cap the size of the returned result set.

— Names exact query functions but not limit syntax or default. v1.16.1

02 **Opt-in Git-triggered index refresh** NEW 35

Adds an opt-in feature that automatically re-indexes the codebase when Git activity is detected, rather than requiring a manual refresh.

— Describes behavior but no config key or trigger details given. v1.16.0

WAS THIS USEFUL? 👍 👎

kane-cli

Feedback ⓘ 0.9.0 NOTES ↗ RANK

Kane CLI by TestMu AI (Formerly LambdaTest)

> Kane CLI 0.8.22 adds a `config reset-url` subcommand, brings mobile execution much closer to web replay parity (including `prefer_intent` support and shared binding logs), and tightens `until` export lowering, imported-file variable scoping, and the `balance` command's reporting.

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

01 **`kane-cli config reset-url` subcommand** NEW 70

New `kane-cli config reset-url` subcommand clears the configured default URL so the next run picks up a fresh endpoint.

Reset a misconfigured or stale default URL so the next run picks up a fresh endpoint.

```
$ kane-cli config reset-url
```

— Exact runnable command, but no further mechanism given 0.8.22

02 **Mobile executor reaches web replay parity** IMPROVED 70

The mobile executor and replay engine now match web behaviour: it adds `prefer_intent` option support, handles hub settings, `until` loops, and empty steps the same way web replay does, surfaces Appium binding logs in the run log alongside web binding logs, and retains each mobile step's own label without inserting a stray navigate step into recordings.

— Named flag and multiple concrete behaviours, but no commands shown 0.8.22

THINNER COVERAGE BELOW

03 **Platform-aware `until` export lowering** IMPROVED 55

`until` inside a taken branch now exports as a loop within that branch instead of being hoisted out, and the export oracle lowers an `until` op using the tape's platform for platform-aware lowering.

— Names the mechanism and op but no user-facing command 0.8.22

04 **`kane-cli balance` reports available credits only** IMPROVED 45

Clarifies `kane-cli balance` to report available credits only, removing ambiguity about what the figure represents.

— Named command with a clear behaviour change, little else 0.8.22

05 **Imported file inherits importer's variables** IMPROVED

25

An imported file now sees its importer's variables, preventing unresolved references.

— States the fix but no mechanism or surface named 0.8.22

NEW SUPPORTED PLATFORM Cloud real-device support for Android and iOS

NEW SUPPORTED PLATFORM Windows smart variable timezone resolution

0.9.0 NOTES ↗ LATEST STABLE

kane-cli 0.9.0 adds cloud real-device runs via `--target android` / `--target ios`, runtime resolution of `{{global.…}}` and `{{smart.…}}` variables, `{{file.<alias>}}` upload references, and a per-objective `max_steps` cap.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch 0.9.0 https://github.com/LambdaTest/kane-cli.git
# already have the repo? check out this version:
$ git checkout 0.9.0
```

Adds `--remote` flag on `kane-cli run` with sensible device defaults for real-device runs from any machine, no local device setup required.

Adds `{{file.<alias>}}` variable syntax to reference uploaded files by alias throughout a test session, with aliases reserved up front to prevent mid-run collisions.

Adds `max_steps` per-objective cap in testmd, with a defined fallback when not set.

Adds a live device window in the TUI showing the real-device session as it happens.

Resolves `{{global.…}}` and `{{smart.…}}` variable templates at runtime when an objective starts, so agents receive real values instead of raw placeholders.

+2 more

WAS THIS USEFUL? 👍 👎

gentle-ai

Feedback ⓘ v4.2.0 NOTES ↗ RANK

Configures AI coding agents like Claude Code, Cursor, and OpenCode with persistent memory, curated skills, MCP servers, and personas. Open source with no agent lock-in.

> Gentle-AI's v4.1.0–v4.2.0 window adds agent-driven review lens selection, a TUI installer for Gentle Shell, an opt-in modular Claude Code orchestrator, and risk-gated native review triggering, alongside provider-contract bundling of ODD routing assets and tighter ODD delegation rules.

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

01 **Agent-selectable review lenses** NEW 85

Adds `--lenses <r,...>` and `--lenses-reason <text>` flags to `review` START and STATUS preflight commands, letting the agent name exactly which 4R lenses to run instead of accepting the tier default. A new `start_lens_selection` capability flag lets callers detect whether the connected binary supports agent-driven lens selection. Provider contract bumped from `1.2.0` to `1.3.0`: review authority carrying a lens selection cannot be read by binaries older than v4.1.0.

— Exact flags and capability flag with version compatibility noted v4.1.0

02 **Claude orchestrator modular install pilot** NEW 81

Adds `gentle-ai install --claude-orchestrator-modules` opt-in pilot to install the global Claude Code orchestrator as a core prompt plus on-demand modules; requires `--scope global` and the `claude-code` agent.

— Exact command with required flags and agent target v4.1.0

03 **Gentle Shell TUI installer** NEW 81

Adds `gentle-ai shell install` with a TUI and an inspect-then-confirm command line for installing Gentle Shell on Linux amd64 (Separate and Shared modes), Windows 11 x64, and macOS (user installer).

— Exact command and supported platforms listed v4.1.0

04 **Risk-gated native review triggering** BREAKING 78

Native review now triggers automatically only for `high-risk` changes; `review assess` reports `review_due: true` only for `high` risk, while a `medium` candidate now reports `review_due: false` with `under_budget` regardless of `changed_lines`. Scripts that relied on `review_due: true` for medium-risk changes will now receive `review_due: false`.

— Exact field values and breaking impact specified v4.1.0

05 **Provider-contract bundling of Gentle Shell ODD routing** NEW

75

Publishes `routing/pi.md` in the provider-contract bundle under a new `routing` manifest field, introduced in contract `1.4.0`, letting agents vendor Gentle Shell's ODD routing block via SHA lock instead of hand-copying it.

Upgrade and resync all managed agent, reviewer, and runtime assets (including the new `routing/pi.md` block) in one step.

```
$ brew upgrade gentle-ai && gentle-ai sync
```

— Names config field and contract version, includes sync command v4.2.0

06 **Install and restore command validation** IMPROVED

60

`install` now rejects component modifiers whose component is not selected, before any self-update. `restore` now honors explicit boolean flag values and the `--` terminator, and rejects extra targets.

— Names exact commands and validation behavior, no example given v4.1.0

THINNER COVERAGE BELOW

07 **ODD orchestration slicing and delegation criteria** IMPROVED

55

ODD routing now commits work in reviewable slices of about 400 authored changed lines on the feature branch, keeping small tasks inline without feature documents or delegated writers unless explicitly requested. ODD orchestration now delegates only when a named reason applies — exploration budget, parallel independent writers, or independent verifier — rather than on file count or task size alone.

— Describes behavior change but no command or flag given v4.2.0 v4.1.0

08 **OpenCode integration restrictions** BREAKING

50

OpenCode external community plugins are retired: `gentle-ai` no longer installs external community plugins or shows their TUI step. The OpenCode orchestrator is now restricted to Gentle AI subagents, and OpenCode permissions now follow the Pi safety model.

— Named restrictions but no migration command given v4.1.0

└──▷ BREAKING ON UPGRADE

! Run `gentle-ai sync` after upgrading — installed agent guidance, orchestrator instructions, and ODD routing assets changed and must match the installed binary.

! OpenCode external community plugins are retired: `gentle-ai` no longer installs external community plugins or shows their TUI step.

! Provider contract bumped from `1.2.0` to `1.3.0`: review authority that carries a lens selection cannot be read by binaries older than v4.1.0 — continue any such lineage with v4.1.0 or later.

! `review assess` no longer marks `medium` candidates as due; scripts that relied on `review_due: true` for medium-risk changes will now receive `review_due: false` with `under_budget`.

! `install` now rejects component modifiers whose component is not selected, before any self-update.

! `restore` now honors explicit boolean flag values and the `--` terminator, and rejects extra targets.

! The OpenCode orchestrator is restricted to Gentle AI subagents, and OpenCode permissions now follow the Pi safety model.

! Run `gentle-ai sync` after upgrading: installed orchestrator prompts, ODD guidance, permissions, reviewer assets, and SDD retirement cleanup all changed.

WAS THIS USEFUL? 👍 👎

hax

Feedback ⓘ v0.6.1 NOTES ↗ RANK

A minimalist, terminal-native coding agent written in C.

> hax v0.6.1 adds a prebuilt macOS Apple Silicon binary and significantly shrinks the Linux binary size.

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

01 **Smaller Linux binary via trimmed libcurl** IMPROVED 55

Prebuilt Linux binary size is reduced to approximately 2 MB, a quarter of the previous size, achieved by bundling a trimmed libcurl.

— Gives concrete size and cause but no action needed v0.6.1

02 **Prebuilt macOS Apple Silicon binary** NEW 50

A prebuilt macOS binary for Apple Silicon is now provided, requiring only libraries shipped with macOS and no Homebrew dependency.

— Names platform and dependency behavior but no install command v0.6.1

WAS THIS USEFUL? 👍 👎

rtk

Feedback ⓘ v1.0.0 NOTES ↗ RANK

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

> rtk v1.0.0 ships a new pipeline engine backing the CLI (`rtk bypass` , `rtk llm-proxy`) and a request-layer `rtk-proxy` daemon, alongside several breaking changes (flag removal, metric redefinition, licence split); v0.52.0 adds fish shell script rewriting and platform hook support.

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

01 **rtk bypass replaces rtk proxy** BREAKING 70

`rtk proxy <cmd>` is renamed to `rtk bypass <cmd>` (the old name is kept as an alias so existing callers don't break immediately), and the renamed command gains CLI pipeline support and a hook gate.

– Exact old/new command names and mechanism given; migration timing unclear v1.0.0

02 **Pipeline engine, llm-proxy command and rtk-proxy daemon** NEW 60

v1.0.0 introduces a pipeline engine built on plugin and text crates for composable request processing, and agent plugins now honour pipeline `block` and `route` directives. The new `rtk llm-proxy` command exposes LLM proxy operations from the CLI, while a separate `rtk-proxy` daemon enforces these pipelines at the request layer.

– Names the commands and directives but not their options or configuration v1.0.0

THINNER COVERAGE BELOW

03 **Fish shell -c script rewriting** NEW 50

rtk now reads a quoted `fish -c` script as fish and rewrites its inner commands through the rewriter.

– Names the exact shell invocation handled but not rewrite scope v0.52.0

04 **Licence split with rtk-oss edition** BREAKING 45

The default binary's licence changes, and an Apache-2.0 edition called `rtk-oss` now ships alongside the default binary.

– Names the new edition but not the new default licence terms v1.0.0

05 **rtk gain metric redefinition** BREAKING 40

`rtk gain` now excludes oversized outputs from its headline rate, which changes previously reported metrics.

— Explains the behaviour change but not the new calculation `v1.0.0`

06 **`--ultra-compact` flag removed** BREAKING 35

The `--ultra-compact` flag is removed with no replacement.

— States the removal but gives no migration path `v1.0.0`

07 **OpenCode V2, Desktop and Windows hook support** NEW 15

v0.52.0 adds hook support for OpenCode V2, Desktop, and Windows, per the release summary.

— Mentioned only in summary with no mechanism or usage detail `v0.52.0`

╰──▷ BREAKING ON UPGRADE

! `rtk proxy <cmd>` is renamed to `rtk bypass <cmd>` (old name remains as an alias — no immediate break, but callers should migrate).

! The `--ultra-compact` flag is removed with no replacement.

! `rtk gain` now excludes oversized outputs from its headline rate, changing previously reported metrics.

! The default binary's licence changes; an Apache-2.0 edition (`rtk-oss`) ships alongside the default binary.

WAS THIS USEFUL? 👍 👎

kooky

Feedback ⓘ v0.51.12 NOTES ↗ RANK

A minimal modern terminal for AI coding experience — Sidebar workspaces; horizontal / vertical split panes; one-click agent launch; per-agent activity readout; live workspace state with one-click Node and branch switching.

> kooky's single release this window focused on keyboard-driven navigation, adding Ghostty/Warp-style shortcuts for moving between splits, cycling workspaces, and jumping to the last tab, plus a non-destructive way to hide sessions in the History panel.

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

01 **Split-pane focus and resize shortcuts** NEW 70

Ghostty/Warp-compatible keyboard shortcuts let you navigate and resize split panes without the mouse: ⌥⌘ + ← / → / ↑ / ↓ moves focus to the split in that direction, and ^⌘ + ← / → resizes the nearest divider.

Navigate between split panes without touching the mouse during an active investigation across multiple terminal sessions.

📍 Press ⌥⌘ + ← / → / ↑ / ↓ to move focus to the split in that direction. Press ^⌘ + ← / → to resize the nearest divider.

— Exact key combos given but no deeper mechanism beyond the shortcuts themselves.

v0.51.12

02 **Workspace cycling shortcut in sidebar** NEW 70

^⌘[/ ^⌘] step to the previous/next workspace with wrap-around, auto-scrolling the sidebar to the landed workspace and expanding its group if it was collapsed.

— Names exact shortcut and describes auto-scroll/expand behavior precisely. v0.51.12

THINNER COVERAGE BELOW

03 **Session hiding in History panel** NEW 55

Sessions can now be hidden from the History panel: hover a row and click the eye icon to hide it, then use show hidden to restore hidden sessions. Only the visibility state is stored — the underlying conversation data is never modified.

Keep the History panel uncluttered by hiding sessions you no longer want to surface, without deleting any conversation data.

📍 In the History panel, hover over a session row and click the eye icon to hide it. To restore hidden sessions, click show hidden.

— Clear mechanism and UI path, but no config key or command given. v0.51.12

04 **Jump-to-last-tab shortcut** IMPROVED 45

⌘9 now jumps to the last tab, matching Ghostty, Warp, and browser behavior.

— Single-line addition with an exact shortcut but no further detail. v0.51.12

WAS THIS USEFUL? 👍 👎

tty7

Feedback ⓘ v26.10.0 NOTES ↗ RANK

The terminal that outlives its window. Shells survive quitting the app, agents can drive other agents, and remote machines work the same way.

> tty7 shipped a mobile companion app for watching and driving panes and agents remotely (iOS and Android), alongside new Windows platform support with remote workspaces and native ARM64 packages, a message composer for agent CLIs, and a handful of smaller UI and localization additions.

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

01 **Phone companion app for panes and agents** NEW 85

A mobile companion app lets you watch and drive your computer's panes and agents remotely, available for iOS via TestFlight and for Android (arm64) via `tty7-<version>-android-arm64.apk` downloaded from the `mobile-v*` releases page. Pair by going to Settings → Mobile → Allow phone access → Show code on the desktop, then scanning or pasting the code in the phone app; installing a newer Android APK over a previous version preserves paired machines.

Pair your phone to your tty7 desktop instance so you can watch and drive panes and agents on the go.

> 📍 On the computer, go to Settings → Mobile, turn on Allow phone access, then click Show code. On your phone, open the tty7 app and scan the displayed code or paste it to complete pairing.

Install the tty7 Android app on an arm64 phone when you do not have access to a managed app store.

> 📍 Download `tty7-<version>-android-arm64.apk` from the latest `mobile-v*` release at `https://github.com/l0ng-ai/tty7/releases?q=mobile-v&expanded=true`, then open the file. Allow installing from your browser or file manager when Android prompts, then complete installation.

Pair your phone to a tty7 desktop session so you can monitor and drive panes and agents remotely.

> 📍 On the computer: Settings → Mobile → turn on Allow phone access → click Show code. On the phone: open the tty7 app and scan or paste the code shown.

Pair your phone to your computer so you can monitor and drive tty7 panes and agents remotely.

📍 On the computer, go to Settings → Mobile, enable `Allow phone access`, then click `Show code`. On the phone, open the tty7 app and scan the code or paste it to complete pairing.

Pair your phone so you can watch and control terminal panes and agents from anywhere.

📍 On the computer: go to Settings → Mobile, turn on `Allow phone access`, then click `Show code`. On your phone: open the tty7 app and scan the QR code or paste it to complete pairing.

— Full pairing flow, platforms and APK naming given `v26.10.0` `mobile-v0.1.6` `mobile-v0.1.5` `mobile-v0.1.2` `mobile-v0.1.1`

THINNER COVERAGE BELOW

02 **Windows remote workspaces and ARM64 packages** NEW — 50

Adds Windows platform support, including Windows remote workspaces and native Windows ARM64 packages for Snapdragon and ARM64 Windows devices.

— Names platforms but no install steps `v26.10.0`

03 **Message composer for agent CLIs** NEW — 30

Adds a message composer for agent CLIs, enabling structured input to AI agent terminals.

— No UI path or mechanism described `v26.10.0`

04 **Process resource usage in UI** NEW — 25

Adds process resource usage display in the UI.

— States feature but no location or metrics `v26.10.0`

05 **Recently closed tabs and close confirmation** NEW — 25

Adds recently closed tabs list and close confirmation.

— Feature named, no UI path given `v26.10.0`

06 **Agent-ready worktree setup** NEW — 20

Adds worktree setup that can start ready for an agent.

— Bare description, no mechanism given `v26.10.0`

07 **Terminal theme and notification protocol support** NEW — 15

Adds terminal theme and notification protocol support.

— No protocol names or config details given v26.10.0

08 **Russian translation** NEW 10

Adds Russian translation.

— Minimal description of localization addition v26.10.0

09 **Additional agent integrations** NEW 5

Adds more agent integrations.

— No specific integrations or names given v26.10.0

NEW INTEGRATION Qoder CN CLI support

NEW INTEGRATION Prime Agent and Empryo agent recognition

NEW INTEGRATION GitHub tab for issues, pull requests, checks, and reviews

v26.9.4 NOTES ↗ STABLE

tty7 v26.9.4 brings language servers, Search Everywhere, scriptable `tty7 exec` / `tty7 send` CLI, sleep-able tabs, and remote file upload via drag-and-drop.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v26.9.4 https://github.com/l0ng-ai/tty7.git
# already have the repo? check out this version:
$ git checkout v26.9.4
```

Saved SSH hosts now live in `servers.json`, giving practitioners a file they can version-control or deploy programmatically.

`tty7 exec` runs a command in an existing pane and returns its result, enabling scripted interaction with live sessions.

`tty7 send` reads from stdin or a file and can paste into a pane, enabling automated input injection.

Adds language-server support to the built-in code editor: diagnostics, hover, completion, and go-to-definition.

Adds multiple cursors, git change markers, a symbol outline, and split view to the built-in code editor.

+9 more

└──▷ BREAKING ON UPGRADE

! Editing a saved host's authentication, proxy, forwarding, and advanced options in Settings has been removed; these options are no longer accessible from the

Settings UI.

- ! The `By repo or folder` grouping mode and `Group Automatically` setting have been removed; the sidebar now groups tabs by repo automatically.
- ! The `Window & Tabs` settings page has been removed (folded into General).
- ! The `Open diff preview from sidebar counts` setting has been removed.
- ! Saved SSH hosts have moved to `servers.json`; any tooling or scripts reading hosts from a previous location must be updated.

WAS THIS USEFUL? 👍 👎

repoprompt-ce

Feedback ⓘ v1.6.0 NOTES ↗ RANK

Community edition of RepoPrompt: a native macOS context engineering app for AI coding agents, with an MCP CLI.

> RepoPrompt CE v1.6.0 adds opt-in Codex Computer Use, new GPT-6.1 Sol and Claude Sonnet 5.5 models with usage-balanced routing, lane-aware Overseer session management, and Code Maps support for non-Git folders.

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

01 **Codex Computer Use for local tab interaction** NEW 70

Adds opt-in computer interaction in Codex chats via Codex Computer Use, which lets the agent interact with local browser tabs after the user grants local tab consent in a dialog; companion activity is quieter while this is active.

Enable Codex Computer Use when you want the agent to interact with local browser tabs on your behalf during a Codex chat session.

📍 In a Codex chat, open the session settings and opt in to Codex Computer Use. When prompted, grant local tab consent in the dialog that appears.

— Names the feature and consent flow, with concrete opt-in steps given v1.6.0

02 **Code Maps for non-Git folders** NEW 60

Code Maps now works against non-Git project directories, providing structural code navigation without requiring a Git repository to be present.

Run Code Maps against a non-Git project directory to get structural code navigation without needing a Git repository.

📍 Open RepoPrompt CE, select a non-Git folder as your project root, and open the Code Maps panel — indexing will proceed without a Git repository present.

— Clear UI navigation path given though limited mechanism detail v1.6.0

THINNER COVERAGE BELOW

03 **New models and usage-balanced routing** NEW 55

Adds `GPT-6.1 Sol` and `Claude Sonnet 5.5` as selectable models, refreshes Cursor models and effort choices, and adds usage-balanced routing for Codex and Claude plans.

— Names specific models but no selection or routing steps given v1.6.0

04 **Lane-aware Overseer session management** IMPROVED 55

Opt-in Overseer management gains lane support, role-aware discovery, and stronger recovery after relaunch or compaction. It also adds a way to communicate with in-app Overseers without taking them over, alongside safer MCP bootstrap handshakes and clearer startup binding results.

— Describes several mechanism changes but gives no concrete usage steps v1.6.0

WAS THIS USEFUL? 👍 👎

zerostack

Feedback ⓘ v1.8.5 NOTES ↗

Lightweight coding agent written in Rust, optimized for memory footprint and performance

Adds `--disable-lazygit` CLI flag, permission-wait status signals (protocol v1.1), and richer TUI input editing (Ctrl+Left/Right, paste, selection).

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.8.5 https://github.com/gi-
dellav/zerostack.git
# already have the repo? check out this version:
$ git checkout v1.8.5
```

Adds `--disable-lazygit` flag (also bound to Ctrl+L shortcut) for use with `multistack` to disable lazygit integration.

Reports permission-wait events over the status-signals Unix socket (protocol v1.1), enabling external status bars to surface when the agent is blocked waiting for user approval.

Adds Ctrl+Left, Ctrl+Right, paste, and selection support in the TUI input field.

NEW INTEGRATION opencode-zen and opencode-go providers

NEW SUPPORTED PLATFORM darcs commands in permissions system

└──▷ BREAKING ON UPGRADE

! The `zerostack-lite-[target]` family of binary builds now ships with `--default-features` instead of `--no-default-features`; any setup that relied on lite builds having no default features compiled in will now include them.

! The `/loop` command is no longer part of the default compile-time feature set; it remains available until the next major release only if explicitly enabled as a feature.

WAS THIS USEFUL? 👍 👎

◆ Model Training & Fine-Tuning

fizgig

Feedback ⓘ v8.9.6 NOTES ↗ RANK

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

> Fizgig's biggest shift this window is becoming a full web app with phone/browser remote access, multi-GPU distributed training, and cloud pod management (Helpers/Pods tabs, Vast.ai), alongside a new multi-subject LoRA Isolation training mode, full Flux 3 Image Klein support, and a long run of MOSS-Image, Studio, and captioning refinements.

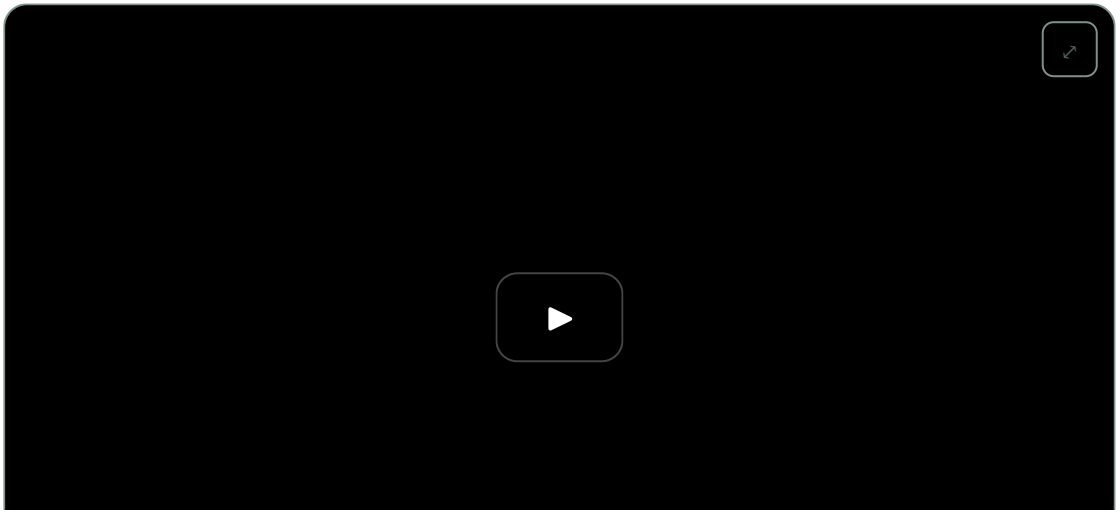
└──▷ WHAT SHIPPED · 36 FEATURES most completely described first ? what's the number?

01 Web app conversion with phone and browser remote access

95

BREAKING

Fizgig now runs as a web app in its own window or any browser, launched via `Launch Fizgig.bat` (Windows) or `./launch_fizgig.sh` (Linux), plus `Launch Fizgig in browser.bat`, `Launch Fizgig AMD.bat`, and `Launch Gizmo.bat`. All user data (settings, presets, cache, downloaded models) is consolidated into a single `user_data` folder with automatic migration. `Preferences → Web UI` lets you set a password and press `Restart Fizgig now` to expose Fizgig on the network, with `Allow on my network` (Windows) opening the firewall on private networks. An `Open on your phone` card on the Start tab shows a QR code (30-second validity, single use) that logs a phone in for 30 days, with foldable tab navigation and a condensed top bar for phone screens. `Preferences → Network` adds `Find Fizgig on my other computers` and `Check for new releases` switches. Breaking: the classic desktop app, `LoRA the Explorer`, and `Diff to LoRA` are removed (covered by Studio's `I'm feeling lucky` and Extract's `From a fine-tune`); `run_fizgig.bat` is renamed `Launch Fizgig.bat`; RunPod pods no longer run a remote desktop, serving the UI on HTTP port 8081 instead of VNC port 6080.





Enable remote access from a phone or second machine so you can watch a training run from anywhere on your network.

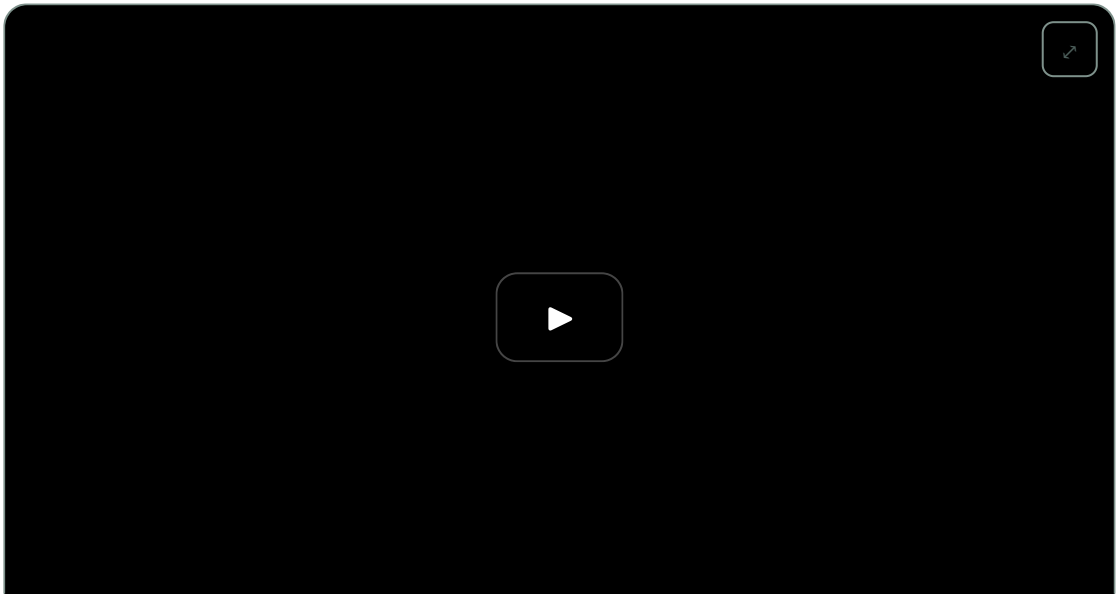
📍 1. Open Preferences → Web UI 2. Set a password in the password field 3. Press Restart Fizgig now 4. Note the network address shown on the card 5. On the Start tab, press Open on your phone and scan the QR code to log in for 30 days

— Names exact launchers, config paths and firewall steps v8.0.2 v8.0.1 v8.9.5

02 **Remote and cloud GPU management via Helpers and Pods tabs** NEW

90

A Pods tab finds other computers on the local network automatically (shown On your network with name/GPU) for use as a remote GPU, and integrates with a RunPod API key to Start / Stop pods, list rentable GPU cards filtered by memory/cloud/price with estimated run cost, and open RunPod's deploy page pre-selected via Deploy ; it was later refined to list only currently rentable RunPod cards sorted cheapest-per-run-first with live stock and pricing. A Helpers tab lists networked computers and running pods with one-click Use it as a helper to enlist a second GPU for training or preview, preview offloading to a helper so training never pauses, and Help another Fizgig with this GPU for a home GPU to help a cloud pod. Adds experimental distributed training across two GPUs, each training its own LoRA copy on a partition of images and merging every 3 epochs by default (merge-every-epoch on fast networks), automatic checkpoint sync from pod to local machine per epoch with epoch-level selection and LoRA Royale evaluation while the pod keeps training, automatic pod deletion once LoRAs are downloaded and verified, Vast.ai support with its own Fizgig template and HTTPS addresses, the `FIZGIG_PASSWORD` template variable for auto-login on RunPod pods, and a Stop this pod when a run finishes option with a cancellable two-minute countdown.



Use a desktop GPU from a laptop on the same network — no configuration needed beyond enabling network access once.

📍 # On the desktop (Windows): open Preferences → Web UI, click Allow on my network, then Restart Fizgig now. # On the laptop: open Fizgig, go to the Pods tab — the desktop appears automatically. # Click Open to train on the desktop's GPU from the laptop's browser.

Deploy a password-protected Fizgig pod on RunPod that logs you in automatically when you open it.

📍 # In the RunPod template environment variables, set: `FIZGIG_PASSWORD=mysecretpassword` # Deploy from: `https://console.runpod.io/deploy?type=GPU&gpu=RTX+5090&count=1&template=bdrdew7hxw` # Open the pod's HTTP port 8081 in your browser — Fizgig loads already logged in.

Set a fixed password for a RunPod pod so it opens already logged in without prompting.

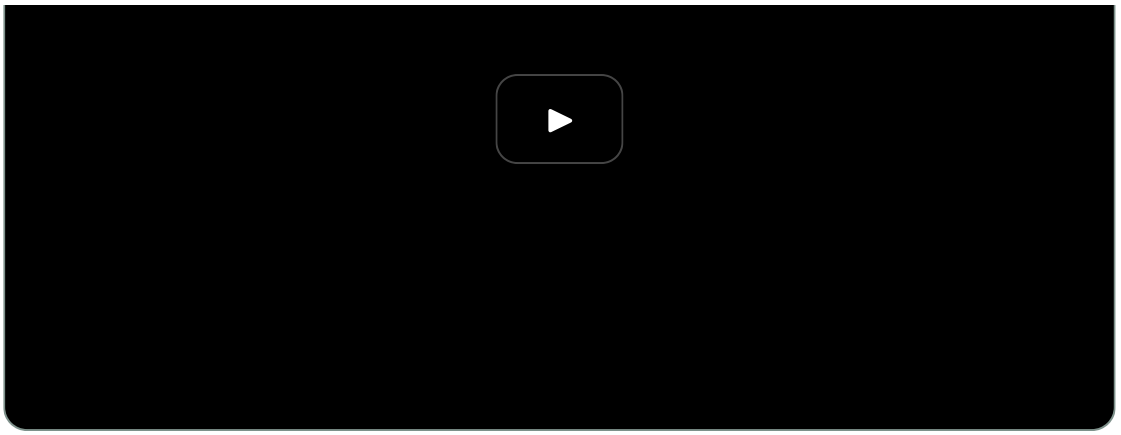
```env
FIZGIG_PASSWORD=mysecretpassword
```

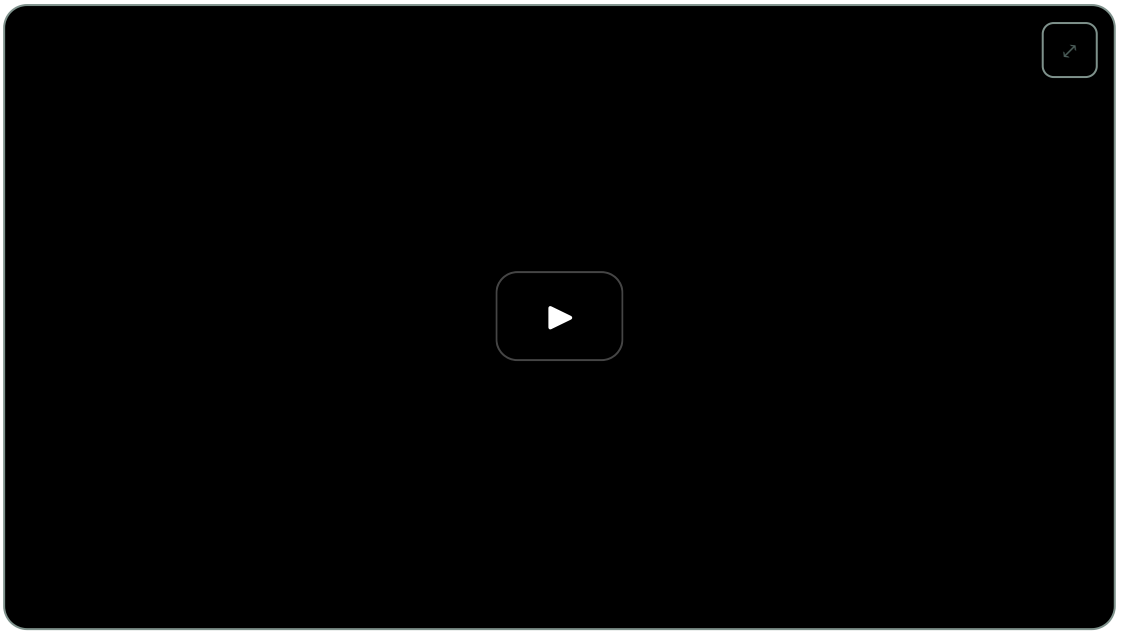
— Names tabs, env var and provider but ranking logic unspecified v8.9.0 v8.0.2 v8.0.0

03 **Multi-subject LoRA Isolation training mode** NEW 90

Fizgig LoRA Isolation trains up to 4 subjects into one LoRA via a Subjects card (each with its own photo folder, trigger word, and type: person, animal/thing, or style), using DOP (Differential Output Preservation), multi-subject identity anchoring, fused-token class decoupling, in-frame negative-pair compositing (with 32 generated reference strangers, 21 MB, downloaded with models), per-sample LR scaling, strided DOP, and balanced sampling to stop identities bleeding onto strangers — reducing face contamination from 48/50 to 0/50 in testing on Krea 2. Also adds Multi Concept (2–4 subject folders) to every model's Standard LoRA and fine-tune, and class-aware training recipes per subject type. Requires two or more subjects — a single-subject form is auto-migrated to Leave the rest of the model alone — and side-by-side comparison images can now prioritise Image Prep's face crops via a `Side-by-side pictures: face crops first` tick.







— Mechanism, measured before/after, and migration path all specified v8.4.0 v8.7.8 v8.9.5

04 **Move to… live run migration across machines and LoRA types** NEW

85

Move to… on the run card migrates a live run to another PC, laptop, or pod, with Carry on here to reclaim it from the last save if the remote disappears. Extended to standard, edit, and slider LoRAs including multi-subject and Fizgig LoRA Isolation runs, relocating associated folders (Subjects 2–4, edit run originals/extras, slider opposite end) and named files (edit run test photo, Samples tab reference picture, thumbnail), with caption edits synced to each folder; runs now also carry their clips and songs, not just pictures. Breaking: moving a run requires the same Fizgig version on both computers.

— Lists every relocated folder/file type and version constraint v8.9.0 v8.9.2 v8.9.3

05 **Flux 3 Image Klein training support** NEW

85

Adds Flux 3 Image Klein (Black Forest Labs' 5.6B text-to-image and editing model) as a trainable model via the `flux-3-image-klein-comfy` Hugging Face repo, supporting LoRA and LoKR training with seven presets (Ultra Fast, Standard, Style, Edit, Edit Strong,

Slider, and Fine-tune), including Edit LoRA training from before/after photo pairs, Slider LoRA training from photo pairs or three prompts, and full base-model fine-tuning. Adds memory-adaptive precision via Auto mode (bf16, INT8, or 4-bit based on available VRAM) with GPU offload during preview renders on small cards; trained LoRAs are compatible with ComfyUI's standard LoRA loader. Adds a reference driver guide entry and extends the driver checklist to cover LoRA Royale's seed and prompt travel.

— Names repo, preset list and precision modes precisely `v8.1.0`

---

06 **Per-image loss tracking in Runs & outputs and run card** NEW

80

Runs & outputs gives a unified view of every run with its loss curve, output files, and per-image loss charts for models with the loss watch, with captions fixable from a chart while the run is live. Extended with a dedicated Per-image loss button on the run card (enabled via Track how each image trains in Training Parameters) showing each image's thumbnail, loss trend, verdict, and editable caption re-encoded at the next epoch; `Show: Worth a look` surfaces persistently hard images; works on all models including MOSS-Image.

— Names exact toggles and verdict/filter controls `v8.0.4` `v8.8.3`

---

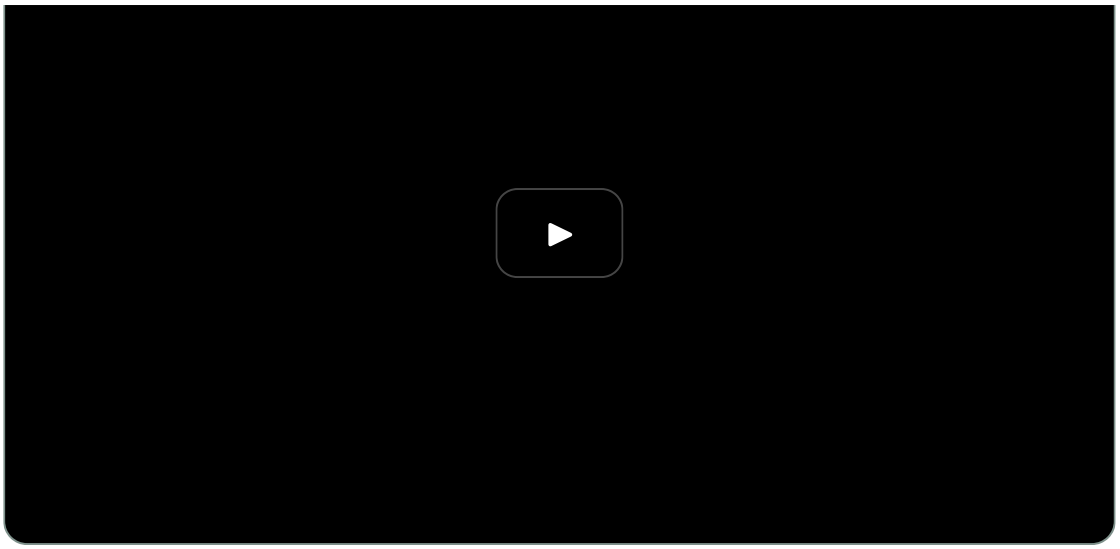
07 **MOSS-Image Training and Preview Improvements**

UPDATED SUPPORTED PLATFORM NEW SUPPORTED PLATFORM IMPROVED

75

For MOSS-Image: a Use the SFT model for samples toggle (Samples tab) renders Base previews through a downloadable SFT-look LoRA (~2.1 GB, auto-downloaded) with no model swap, with the 8-step Turbo LoRA still available by unticking it and setting `Turbo LoRA strength` to 1 (12 steps, CFG 1, activatable mid-run), each enlarged gallery preview now showing its render-provenance (SFT, Base + SFT-look LoRA, or Base alone, plus Turbo status) and exact prompt, and a `Render with:` control adding Use MOSS-Image SFT and Follow Samples tab settings options shared across Studio, LoRA Royale, and the Profiler; a Clean-end share option (Training tab) defaults to 60% to favor low-noise steps (sigma below 0.5) with a new Slide mode as the default for all presets (ramping 60% through epoch 7, tapering to 30% by epoch 15, then returning to 60% every 2nd epoch), and EMA is removed from the Style preset since a style LoRA at 5e-5 converges slowly enough without it; a `1.05` Target Megapixels option trains MOSS-Image at exactly 1024×1024 (up from the nearest 1.0 at 992×992), and Block Swap can now be changed while a run is paused and takes effect on Resume without restarting; the training tick 'Base is SFT or a fine-tune of it' is renamed Train attention only, restricting training to attention layers and cutting MOSS-Image fine-tune peak VRAM to 12.6 GB so it fits whole-model on a 16 GB card with no rotation; and the text encoder now stays in RAM between prompts, cutting render time from ~37s to ~25s on 32 GB cards.





– Named toggles and exact step/CFG values given Exact percentages and epoch schedule given, no CLI step Exact resolution figures given, no further mechanism Concrete VRAM figure, single checkbox rename Concrete timing numbers but fully automatic, no user action

v8.9.3 v8.8.1 v8.7.7 v8.7.3 v8.7.0 v8.7.1 v8.7.5 v8.5.9

---

08 **Two-stage preview and checkpoint save scheduling** IMPROVED

75

Replaces the former Then preview less often control (itself new, with After epoch / Preview every N epochs sub-fields) with a two-stage Every N epochs / `then from epoch` Y `every` Z epochs preview schedule on the Samples tab, supporting both fewer and more previews later. Adds an equivalent two-stage checkpoint save schedule on the Training tab; the last epoch is always saved, and both Resume and 'Change the previews of the run going now' pick up live changes.

– Exact field names and behaviour on Resume given v8.7.7 v8.7.8

---

09 **Studio 'I'm feeling lucky' roll panel and 'Keep in view' previews**

NEW

75

Adds an I'm feeling lucky panel to Studio with a configurable Roll control (1, 2, 4, up to the model's block count, or All blocks), a Freeze tweaked toggle locking hand-moved/rolled blocks with an Unfreeze action, a free-slider count display, a Slider strengths from ... to ... range control, and donor targeting choosing whether a roll changes the primary, donor, or both. Later adds individual padlocks to every donor slider for independent lock/freeze/roll, and a `Primary / Donor / Both` selector with Roll counting sliders separately. A Keep in view panel floats or pins Baseline and Tweaked previews on-screen with resize, drag-to-float, and pin-to-top-with-adjustable-height; Donor LoRAs mode splits the left pane to show the user's LoRA alone, and the first preview render is reduced to fill both panes when no slider has moved.

– Many named controls but navigation only, no commands v8.2.0 v8.3.0

---

10 **Extra reference pictures for Edit LoRAs** IMPROVED 75

Edit LoRAs on Flux 3 Image Klein, Klein 9B, and Qwen Image 2.1 accept extra reference pictures named with a numeric suffix (e.g. `beach_1.jpg`, `beach_2.jpg` for `beach.jpg`) — up to 9 per photo on Flux 3 Image Klein and Qwen Image 2.1, up to 3 on Klein 9B — via The photo + extra pictures option under What the edit uses on the Training tab's Edit pairs card, with an Extra pictures folder picker. Previews for edit runs use the test photo's own extras (e.g. `test_1.jpg`) or the first original's. Breaking: extra pictures no longer go in the Originals folder as in 8.1.1; they must be moved to their own folder with The photo + extra pictures selected.

— Exact naming scheme, per-model limits, and migration step given v8.1.1 v8.1.2

---

11 **Kromoss removed for MOSS-Image** NEW INTEGRATION BREAKING 70

Kromoss is removed as a selectable preview model and training base for MOSS-Image. Users who had the preview model row or Base DiT row set to Kromoss must go to Preferences → MOSS-Image and press Download models for me to restore correct defaults (SFT on the preview row, MOSS-Image Base on the Base row).

— Exact remediation path given for the breaking change v8.7.7

---

12 **Real-time slider prompt steering mid-training** NEW 70

Adds real-time slider prompt steering via the Training tab's Slider prompts panel: update subject description, each end, and push strength mid-training; changes take effect at the next epoch with no restart, and every change is recorded in the saved LoRA.

— Names the panel and exact live-update behaviour v8.0.4

---

13 **Captions Tab Editing and Compatibility Improvements** IMPROVED 65

On the Captions tab: externally edited captions are now written back in the run's own extension (e.g. `.caption`) in addition to `.txt`, caption files saved in Windows-1252 and UTF-16 encodings are now read correctly instead of emptying their folder, per-picture editing lets you click a picture to open a full view and Save, Re-caption, or Remove its caption with arrow-key navigation that auto-saves edits before stepping (removed pictures move to a `removed images` folder rather than being deleted), and the tab now displays which folder it is captioning, matching Image Prep's behaviour.

— Names formats and extension but limited mechanism detail Exact UI actions and destination folder named

v8.9.0 v8.9.3 v8.0.4 v8.0.6

---

14 **Learning Rate controls: live resume changes and reset** IMPROVED 65

Learning Rate (and Min/Max when Adaptive LR is on) can be changed before pressing `Resume`, with the run continuing at the new rate and the console logging the change. A `Reset to 1e-5 / 4e-4` action restores `Adaptive LR` defaults when Min/Max have been altered.

— Exact default values and resume behaviour named v8.8.3 v8.0.6

---

15 **Trigger word checker for training subjects** NEW 65

Adds a trigger word checker to the Training tab's Subjects card (collapsed section) that runs a word through the model's text encoder to show how it splits, flags shared letters or subword pieces with other subjects' triggers, and suggests cleanly-joining alternatives — checked per-subject since a word can join well with one subject type and break with another.

— Explains mechanism and exact UI location v8.4.2

---

16 **RefMod Toolkit for MiniMax H3** NEW 65

Adds `RefMod Toolkit` for MiniMax H3: try RefMods and ReFLoRAs side by side on the same seed, run sweeps of one setting at a time, view the exact ComfyUI node values, and Bake or Save as ReFLoRA. Later adds `Import...` to load a graph preset from any folder, not only from the RefMod folder's `graph_presets` directory.

— Names toolkit functions and import path extension v8.0.4 v8.0.6

---

17 **Qwen Image 2.1 training adapter v2** IMPROVED 65

Upgrades Fizgig's Qwen Image 2.1 training adapter to v2 (~84 MB download), automatically switching Preferences if on v1; the v1 file is kept on disk and also available on Hugging Face at `ShootTheSound/Fizgig-Qwen-Image-2.1-Training-Adapter`.

— Exact repo path and file size, automatic switch v7.3.1

---

18 **Gradient checkpointing auto-management on large GPUs**

NEW SUPPORTED PLATFORM IMPROVED 65

Automatically disables gradient checkpointing on 40 GB+ cards (A100, A40, A6000, L40S, H100, RTX PRO 6000) when a run fits with headroom, delivering 13–27% faster training per step on Krea 2, Qwen Image 2.1, MOSS-Image, Z-Image and MiniMax H3; where VRAM is tight, it disables checkpointing only on the last blocks for 2–21% gains.

— Concrete GPU list and performance ranges, fully automatic v8.9.5

---

19 **Anime and furry face detection** NEW 60

Adds anime/furry/drawn-character face detection and likeness scoring using an anime/furry head model and DINOv2, selectable per dataset or automatic, alongside existing InsightFace and ArcFace used for people.

— Names the models used but no usage walkthrough v8.9.0

20 **Low-RAM Krea 2 construction via layer-by-layer quantisation**

IMPROVED 60

Adds layer-by-layer construction and quantisation for Krea 2 when system RAM is insufficient, reducing peak RAM from ~25 GB to ~2 GB with identical weights.

— Concrete before/after numbers, but fully automatic v8.9.0

21 **H3 Speed Boost clip rendering** BREAKING 60

Adds an H3 Speed Boost tick on the Clip row (on by default): renders the first half of steps at half size then grows to full size, speeding up clip rendering based on the SPEED method (Xiao, Chao, Yariv and Wetzstein, 2026). Breaking: H3's Render size menu and No-LoRA clip are removed — clips render at the Width and Height set directly, and the base-model render no longer appears in the loop.

— Names mechanism and cited method, removal path clear v8.3.0

22 **Extract tab for fine-tune to LoRA conversion** NEW 60

Adds an `Extract` tab to convert a full fine-tune into LoRAs at multiple ranks in one pass, replacing the separate Checkpoint to LoRA app; the Fine-tune card now links directly to `Extract > From a fine-tune` for sharing a fine-tune as LoRAs.

Convert a full fine-tune checkpoint into LoRAs at multiple ranks in one pass using the new Extract tab.

📍 # In the Fizgig web UI, open the Extract tab. # Select your .safetensors fine-tune file. # Tick the ranks you want (e.g. 4, 8, 16). # Click Extract — one LoRA per rank is saved to your output folder.

— Runnable example included showing exact tab workflow v8.0.4 v8.0.6

23 **Metadata editing for .safetensors files** NEW 60

Adds `Metadata` editing for `.safetensors` files: read and edit metadata and thumbnail, with header-only rewrites so even large checkpoints save in seconds.

— Names file format and rewrite mechanism, no UI path given v8.0.4

24 **In-app help content expansions** IMPROVED 60

Adds contextual help across panels: the Timestep range panel explains 0-400 trains the clean end (style, texture, fine detail) and 600-1000 the noisy end (composition, layout, pose); the Previews panel gets an overtraining tip (give one prompt something the dataset lacks, keep the last epoch that still follows it); the Fine-tune panel explains Window size and regularisation photos (real photos, 0.1-0.3 for a light anchor); the Studio panel covers Steps/Turbo on H3, editing a Turbo LoRA (set Turbo to 0), greyed-out block rows, and out-of-VRAM troubleshooting; the Royale panel gives reference

strength guidance (about 0.1-0.4 for seed travel, 1.0 for prompt travel) and comparison sheet render cost.

— Specific numeric guidance but it is documentation, not a control v8.0.5

25 **Append speech captions via Whisper** NEW 60

Adds Append speech to the Captions tab: on a clip with sound, Whisper appends what is said to the end of its caption (e.g. `... saying "the words"`), using Gizmo's Whisper model and language setting, on clips and pods.

— Names the control and output format, UI only v8.0.6

26 **Repair Studio visualization and playback controls** NEW 60

Adds block-activity visualisation in Repair Studio for songs (ACE-Step): each block's name glows proportionally to its LoRA influence at the current playhead position, measured during the normal render at no extra cost. Adds playhead scrub handles at the top and bottom of the waveform (Stop becomes Pause during playback, Resume once paused) and draggable edge tabs on the loop section to move either edge independently or expand to the full song.

— Mechanism and UI behaviour described, no exact numbers v7.3.1

THINNER COVERAGE BELOW

27 **Studio Width/Height controls and donor LoRA auto-visibility** IMPROVED 55

Adds separate Width and Height controls in Studio (256–1536) replacing the single square Res box, so previews can match non-square training shapes; values are saved with each setup. Donor LoRAs loaded after the model now immediately display their sliders and buttons without a model reload.

— Exact value range named, UI-only change v8.1.4

28 **Graceful shutdown during training** NEW 55

Quitting during a run now prompts for confirmation, stops the run, records it as stopped, and preserves saved epochs so training can resume from the last checkpoint.

— Clear before/after behaviour, no named config surface v8.0.4

29 **Resilient resume-state saves during training** IMPROVED 50

If writing an epoch's resume state fails (full disk, sync tool, or antivirus interference), Fizgig retries and, if that also fails, warns and continues training rather than ending the run — the epoch's LoRA output is preserved; only the ability to resume from that specific epoch is lost.

— Describes failure/retry behaviour but no config surface v8.5.9

30 **AMD GPU auto-selection for multi-GPU PCs** IMPROVED 50

AMD PCs with both integrated graphics and a discrete GPU now automatically train on the discrete card; the launcher logs its selection and leaves any user-set `HIP_VISIBLE_DEVICES` untouched.

— Names env var but behaviour is automatic v8.1.4

31 **Queue management: Up next panel and Clear queue** NEW 50

Adds an Up next panel in the sidebar on every tab showing the queued run order, with held-queue amber status and a Start queue button, plus a Clear queue button with a confirmation prompt before emptying.

— Named controls but limited mechanism beyond ordering v8.0.4 v8.0.6

32 **Mid-pause target resolution change** IMPROVED 50

Target Megapixels can be altered while a run is paused, and Resume offers to continue at the new size, re-caching only images that no longer fit.

— Describes re-caching behaviour but limited scope detail v8.0.4

33 **EXIF orientation handling across the pipeline** IMPROVED 45

Reads EXIF orientation (rotation flag) correctly across Image Prep, captioning, face finding, training, references and thumbnails, and re-caches previously sideways images upright on the next run.

— Names affected modules but no user-facing control v8.9.5

34 **Mid-run Turbo/Distilled preview toggle** IMPROVED 45

Allows ticking or unticking Qwen's Turbo or Klein's Distilled preview-model option mid-run, taking effect on the next preview rather than requiring a new run.

— Names the two toggles and live-update behaviour v8.9.5

35 **Torch-only attention training** BREAKING 40

Removes the Attention choice from `Training > Other Options`, as every model now renders with torch attention.

— Short note, minimal mechanism or migration detail v8.0.6

36 **'Use that folder…' dataset shortcut** NEW 35

Adds a `Use that folder...` shortcut to an empty training data list for pointing to datasets stored outside the default location.

— Single named shortcut, no further mechanism given `v8.0.3`

└──▷ BREAKING ON UPGRADE

! Moving a run now requires the same version of Fizgig on both computers; mismatched versions will prevent a move from completing.

! Saved forms using Fizgig LoRA Isolation with a single subject are automatically migrated to `Leave the rest of the model alone` on upgrade.

! Kromoss is removed as a selectable preview model and training base for MOSS-Image; users who set the preview model row to Kromoss or the Base DiT row to Kromoss must go to Preferences → MOSS-Image and press `Download models for me` to restore the correct defaults (SFT on the preview row, MOSS-Image Base on the Base row).

! H3's Render size menu and No-LoRA clip are removed: clips now render at the Width and Height you set directly, and the base-model render no longer appears in the loop.

! Extra pictures no longer go in the Originals folder (as they did in 8.1.1): move them to their own separate folder and select `The photo + extra pictures` under `What the edit uses`.

! The launcher `run_fizgig.bat` is renamed to `Launch Fizgig.bat`; any shortcut pointing at the old filename must be updated.

! The launcher `run_fizgig.bat` is renamed to `Launch Fizgig.bat` — any shortcut pointing at the old filename must be updated.

! `run_fizgig.bat` is renamed to `Launch Fizgig.bat`; any shortcut pointing at the old filename must be updated.

! The classic desktop app is removed; `LoRA the Explorer` and `Diff to LoRA` are gone and have no direct replacement (Studio's `I'm feeling lucky` and Extract's `From a fine-tune` cover their use cases).

! On RunPod, the pod no longer runs a remote desktop — port 6080 is gone; the app is now served directly on HTTP port 8081.

! `run_fizgig.bat` is renamed to `Launch Fizgig.bat`; any shortcut pointing at the old file must be updated.

! The classic desktop app is removed; `LoRA the Explorer` and `Diff to LoRA` no longer exist — use Studio's `I'm feeling lucky` and Extract's `From a fine-tune` respectively.

! RunPod pods no longer run a remote desktop; the new template serves the web UI directly on port 8081 with no VNC on port 6080. Existing Classic pods continue to run unchanged, but new deployments must use the new Fizgig template.

`v8.9.6` `NOTES ↗` **LATEST** `STABLE`

Fizgig v8.9.6 introduces masked training, letting practitioners weight specific image regions during LoRA training via paired mask files, with metadata written to the saved LoRA.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v8.9.6
https://github.com/shootthesound/Fizgig.git
# already have the repo? check out this version:
$ git checkout v8.9.6
```

Adds `ss_masked_training`, `ss_mask_floor`, and masked-picture count to saved LoRA metadata so downstream tools can inspect training conditions.

Adds one-click automatic mask generation on the Image Prep tab for eight region presets, running on CPU via Google's person finder (16 MB download on first use).

Adds a subject-isolation mode that uses the Look Filter's 3 baselines to identify a specific person by face when multiple people appear in a training image, treating all others as background.

Adds a How much outside the mask counts control (default 0.1) to set background region gradient contribution, with 0 ignoring it entirely.

Enables mask-aware file operations so Remove, Move Marked, resize, face crop, and training-share moves all carry each picture's `photo-masklabel.png` along automatically.

+1 more

v8.7.4 NOTES ↗ STABLE

Fizgig v8.7.4 adds Kromoss as the MOSS-Image preview model with an 8-step Turbo LoRA, mid-run control over Steps/CFG/Turbo strength, and a redesigned Samples tab.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v8.7.4
https://github.com/shootthesound/Fizgig.git
# already have the repo? check out this version:
$ git checkout v8.7.4
```

Adds `Turbo LoRA strength` box on the Samples tab to enable the Kromoss 8-step Turbo LoRA for fast previews; setting strength to 1 switches to 12 steps and CFG 1, and can be toggled mid-run.

Adds Change the previews of the run going now on the Samples tab to push Steps, `CFG`, and `Turbo LoRA strength` to the active run without pausing, in addition to prompts, seed, and size.

Adds Use the preview model for samples tick on the Samples tab (on by default) to render training previews on Kromoss or SFT instead of Base.

Adds Kromoss (int8, 5.4 GB) as a selectable Preview model (SFT or Kromoss) in `Preferences → MOSS-Image`, downloadable via `Download models for me`, replacing SFT in previews for Repair Studio and LoRA Royale too.

Adds the lodestones 8-step Turbo LoRA for Kromoss (1.2 GB), downloadable from the `Turbo LoRA` row in `Preferences → MOSS-Image`, replacing the earlier SFT Turbo LoRA.

+6 more

└──▷ BREAKING ON UPGRADE

- ! `Clean-end share` is automatically changed to 30 on first open after updating if it was still at 8.7.0's value of 60; any other custom value is preserved.
- ! Turbo preview steps are automatically changed from 8 to 12 on first open of the MOSS-Image Samples tab after updating if the Turbo LoRA was on and steps were set to 8.
- ! After updating, `Use the preview model for samples` is turned off the first time the Samples tab opens if you were training on SFT or a fine-tune, so previews render on Base; re-tick it to restore preview-model previews.

v8.7.2 NOTES ↗ STABLE

Fizgig v8.7.2 extends Clean-end share (30% default, `--family_option lownoise_pct=30`) to MOSS-Image fine-tunes and adds automatic GPU memory rotation for fine-tuning on 10-16 GB cards.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v8.7.2
https://github.com/shootthesound/Fizgig.git
# already have the repo? check out this version:
$ git checkout v8.7.2
```

Adds `--family_option lownoise_pct=30` CLI argument to set MOSS-Image Clean-end share from the command line.

Extends Clean-end share (30% default) to MOSS-Image fine-tunes, not just LoRAs, so fine-tune training steps now spend 30% at the clean end of the noise range where faces and fine detail are refined.

Enables attention-only fine-tuning on MOSS-Image SFT bases when the `Base is SFT or a fine-tune of it` option is ticked, preventing anatomy degradation from training feed-forward layers.

Adds automatic GPU memory planning for MOSS-Image fine-tuning at 1 MP: Fizgig partitions and rotates model parts to fit 10 GB, 12 GB, or 16 GB+ cards, streaming blocks

from system RAM on 10 GB cards.

v8.6.0 NOTES ↗ STABLE

Fizgig v8.6.0 restores the MOSS-Image Fast preset (0.25 MP, rank 16) and extends Standard and Fast presets to 70 epochs for better likeness convergence.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v8.6.0
https://github.com/shootthesound/Fizgig.git
# already have the repo? check out this version:
$ git checkout v8.6.0
```

Restores the MOSS-Image Fast preset (Standard settings at 0.25 MP, rank 16) for quicker runs and smaller LoRA cards.

Extends MOSS-Image Standard and Fast presets to run 70 epochs to accommodate slow convergence on likeness datasets.

v8.5.7 NOTES ↗ STABLE

Fizgig v8.5.x adds MOSS-Image training, a Linux updater script, Qt-in-venv window support, and File picker sorting/filtering.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v8.5.7
https://github.com/shootthesound/Fizgig.git
# already have the repo? check out this version:
$ git checkout v8.5.7
```

Adds `./update_fizgig.sh` Linux updater script; Linux (NVIDIA) installs should run `git pull` then `./update_fizgig.sh` for all future updates.

Adds Qt installation inside Fizgig's venv on Linux so `./launch_fizgig.sh` opens Fizgig in its own window without requiring separate Qt installation.

Adds Name, Date modified, and Size columns to the File picker, with sorting by name, date modified, date created, size, or type and a filter box.

Adds a Turbo LoRA slot in Preferences for MOSS-Image, with a `Turbo LoRA strength` control on the Samples tab and configurable Steps and CFG.

Adds a Target Megapixels control on the Training tab to override the MOSS-Image 1 MP default for smaller cards.

+1 more

└──▷ BREAKING ON UPGRADE

! The MOSS-Image Fast preset (0.25 MP) is removed; workflows relying on it must switch to Standard or lower Target Megapixels manually.

! Linux (NVIDIA) users must run `git pull` then `./update_fizgig.sh` once to adopt the new updater.

! Linux AMD users must re-run `./install_fizgig_rocm.sh` to get the Qt-in-venv window support.

v8.5.2 NOTES ↗ STABLE

Fizgig v8.5.2 adds MOSS-Image training with all LoRA kinds and workbench tools, INT8/NF4 memory optimisation, fixed-LR presets, and an optional Turbo LoRA for previews.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v8.5.2
https://github.com/shootthesound/Fizgig.git
# already have the repo? check out this version:
$ git checkout v8.5.2
```

Trains Standard LoRA, LoKR, sliders, full fine-tune, Multi Concept, DOP, and Fizgig LoRA Isolation on MOSS-Image.

Previews on MOSS-Image SFT while training on Base; a Samples tab toggle switches back to Base.

All workbench tools — Repair Studio with Turbo Preview, LoRA Royale, the Profiler, and Extract — available for MOSS-Image.

Five training presets (Fast, Standard, Style, Slider, Fine-tune) using fixed learning rate, which suits MOSS-Image better than Adaptive LR.

Auto trains on INT8 base (NF4 on smaller cards) with compiled blocks for ~25% faster steps; text encoder streams through system RAM one layer at a time on 8 GB cards.

+2 more

v7.3.0 NOTES ↗ STABLE

Fizgig v7.3.0 adds ACE-Step 1.5 XL music LoRA training, music sliders, full fine-tuning, Checkpoint to LoRA conversion, and Repair Studio as a DAW for AI music with audio-to-audio, Model EQ Mode, Epoch blend, LoRA mix, block locks, Save/Load Setup, and undo/redo.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v7.3.0
https://github.com/shootthesound/Fizgig.git
# already have the repo? check out this version:
$ git checkout v7.3.0
```

Adds LoRA training on songs for ACE-Step 1.5 XL using a folder of songs with caption `.txt`, optional `.lyrics.txt`, and optional `.json` (bpm/key); LoRAs load directly in ComfyUI.

Adds the ✨ ACE-Step Official LoRA (rank 64, 800 epochs) preset, checkpointing and previewing every 40 epochs across a 10-song album.

Adds full fine-tuning for ACE-Step 1.5 XL via the `✨ ACE-Step Fine-tune` preset — 800 rotations, checkpoint and preview every 50, producing ~10 GB complete model files loadable in ComfyUI.

Adds `Checkpoint to LoRA` conversion to turn a full ACE-Step fine-tune checkpoint into a ComfyUI-compatible LoRA.

Adds music slider LoRAs for ACE-Step: a rank-8 dial between two described musical poles (e.g. calm ↔ energetic), trained via the ✨ ACE-Step Slider (rank 8, 20 epochs) preset in ~15 minutes.

+12 more

WAS THIS USEFUL? 👍 👎

AGENT

◆ MCP Servers & Tooling

gflow-cli

Feedback ⓘ v0.90.0 NOTES ↗ RANK

Command-line tool for Google Flow: generate Veo videos and Imagen images via scripts, batches, or pipelines. Includes an MCP server for coding agents.

> gflow-cli shipped batch run resumption, headless auth session renewal, an agent-only composer mode with video extension, nano2.1 model support with new listing surfaces, and environment variables for pinning a default project and routing browser traffic through a proxy.

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

01 **Resume flag for batch runs** NEW 85

`gflow run` gains a `--resume` flag that reuses rows a previous run already created in the same project instead of starting over, avoiding resubmitting work or burning extra generation credits.

Resume a previously interrupted batch run to reuse already-created rows and avoid resubmitting work or burning extra credits.

```
$ gflow run --config examples/sample_config.json --profile
<your-profile> --resume
```

— Flag and reuse mechanism both described with a runnable command v0.90.0

02 **Model catalog support and listing** NEW 85

Adds `nano2.1` to the accepted model identifiers alongside existing models such as `nano2`, surfaces Flow's model names via `gflow models --json`, and exposes the same list to MCP clients through the new `gflow_list_models` tool, described as the MCP twin of `gflow models --json`.

Use the newer `nano2.1` model for a faster, lower-cost text-to-image generation.

```
$ gflow image t2i "a hot air balloon over Tokyo at sunrise" --
model nano2.1
```

— Names model id, CLI command and MCP tool with an example v0.89.0

03 **Headless auth session renewal** NEW 85

Adds a `--refresh` flag to renew expired sessions without re-running the full login flow, and a `gflow auth renew` subcommand for headless session renewal from the terminal, useful in CI or scheduled pipelines.

Renew a stale Flow session non-interactively in a CI or scheduled pipeline without opening a browser login dialog.

```
$ gflow auth renew
```

— Named flag and subcommand with a runnable CI example v0.86.0

04 **New configuration environment variables** NEW 80

Adds `GFLOW_CLI_PROJECT` to pin a default project so commands no longer need `--project` on every invocation, and `GFLOW_CLI_BROWSER_PROXY` to route every browser launch through a proxy (e.g. `GFLOW_CLI_BROWSER_PROXY=http://127.0.0.1:7897`), useful in networks that block Google Flow directly.

Pin a default project so every `gflow` command targets it without repeating `--project`.

```
$ export GFLOW_CLI_PROJECT=my-project-id
gflow video t2v "Slow cinematic push-in on a sunlit forest clearing" --aspect 16:9
```

Route all gflow browser automation through a local proxy (e.g. Clash) before logging in or generating — useful in networks that block Google Flow directly.

```
$ GFLOW_CLI_BROWSER_PROXY=http://127.0.0.1:7897 gflow auth login --browser chrome
```

— Named env var with a full runnable example Named env var, example value and runnable login command

v0.89.0 v0.88.0

05 **Extend existing video clips** NEW 70

Adds `gflow video extend` to continue an existing generated video clip past its current length, using Flow's own UI automation.

Extend an existing generated video clip past its current length using Flow's native UI automation.

```
$ gflow video extend
```

— Named command with an example though mechanism is brief v0.86.0

THINNER COVERAGE BELOW

06 **Documented wait parameter in MCP schema** IMPROVED 40

The MCP tool schema now documents the `wait` parameter so agents reading the schema can discover and use it directly.

— Only a documentation note, no usage example given v0.90.0

07 **Agent-only composer mode for generation** NEW 40

Adds an agent-only composer mode for `text-to-image` and `text-to-video` generation.

— Named modes but no mechanism or example given v0.86.0

08 **Catalogued image reference for migrated projects** NEW 25

Adds support for referencing a catalogued image of the target project in place, for projects that have been migrated.

— Vague description with no named surface or example v0.87.0

WAS THIS USEFUL? 👍 👎

agentdock

Feedback ⓘ v1.1.1 NOTES ↗ RANK

Secure MCP runtime for AI agents to operate local machines, servers, and containers with multi-device orchestration.

> AgentDock v1.1.1 upgrades its Task Progress system to a new live-card tool pair, adds a dedicated pairing channel for NexusDock, and improves default directory resolution for service and tunnel commands.

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

01 **Task Progress live-card v2 tooling** IMPROVED 80

Task Progress is upgraded to live-card v2: new tasks are created via `task_create` and maintained/polled via `task_manage`, replacing the separate `task_snapshot` tool.

– Names exact replacing tool commands, lacks further behavioural detail v1.1.1

02 **Auto-resolved runtime directory for service and tunnel** IMPROVED 65

The `service` and `tunnel` commands now auto-resolve each platform's default runtime directory, while still respecting explicit arguments and environment variable overrides.

– Names exact commands and override behaviour, no example given v1.1.1

THINNER COVERAGE BELOW

03 **Dedicated WebSocket entry for NexusDock pairing** NEW 50

Adds a dedicated WebSocket entry point for NexusDock pairing, with automatic fallback to the original pairing address on network errors; self-hosted services continue using their own configuration.

– Describes fallback mechanism but no endpoint or config name given v1.1.1

WAS THIS USEFUL? 👍 👎

morluto REA

Feedback ⓘ rea-agents-6.8.0 NOTES ↗ RANK

Reverse engineer anything with agents, from app behavior down to native binaries.

> REA's agent fleet expanded sharply across three releases, adding native-binary and capture inspection surfaces (Android/ADB, Ghidra architecture support, SQLite, packet capture, QUIC, Fiddler SAZ, WASM, Frida), new IDA/JEB MCP integrations and RPC discovery, alongside a large batch of breaking schema changes across capture validation, WebMCP, filesystem digests, provider outputs, semantic graphs and Objective-C facts.

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

01 Ghidra architecture and analysis authority expansion

NEW SUPPORTED PLATFORM NEW INTEGRATION IMPROVED 85

Adds `set_address_name` and `set_addresses_names` mutation authority with namespace-aware data labels and per-address batch results, plus configuration for import language/compiler specifications, seed functions, code and labels before auto-analysis, and support for Windows PE DLLs and bounded MIPS ELF analysis including carved MIPS inputs with incomplete section tables. Further adds Ghidra RISC-V support for little-endian RV32/RV64 `ET_EXEC` ELF with the supported double-float ABI; a PSP static ELF profile with bounded little-endian ELF32 `ET_EXEC` and documented PSP/ABI declarations plus a compatible caller-installed Allegrex extension; MIPS32r2/o32 relocatable objects with soft-float integer-register storage, admitted MIPS16 shared ELF, and a narrowly admitted recordless legacy MIPS32 executable profile; raw architecture/entry input and typed incoming references at exact addresses for compatible Linux/macOS Ghidra profiles; and an analysis settings capability to apply validated analyzer overrides before analysis while retaining the selected analysis profile.

— Names exact mutation APIs, architectures and ABI constraints rea-agents-6.5.0 rea-agents-6.4.0

02 Capture and web-diff validation now rejects stale evidence

83

BREAKING

Historical captures without current producer accounting or transaction identity, and web diffs missing accessibility or storage dimensions, are now rejected — recapture with the current producer and update comparison consumers to include every dimension (see `process-capture.md` and `browser-scenario-contract.md`).

— Gives exact rejection conditions and migration docs rea-agents-6.3.0

03 New server configuration and setup options NEW

78

Adds a `REA_MCP_INPUT_SCHEMA_PROFILE=compact` environment variable to reduce MCP schema payload size for size-capped clients such as Kimi/Moonshot while retaining canonical runtime validation; a `finalization_ms` option on process capture to allow shutdown output collection before timeout escalation; Pi, Hermes and Qwen Code LLM provider setup including Hermes active-profile support; and OMP registration during setup.

— Exact environment variable and value a reader can set today Named config option with clear purpose, minimal mechanism Names providers but no setup detail Single bare sentence with no mechanism or scope

`rea-agents-6.4.0` `rea-agents-6.3.0`

---

04 **New artifact and protocol inspection capabilities**

NEW SUPPORTED PLATFORM NEW 76

Adds managed assembly inspection improvements with `target_declaring_type` and `field_declaring_type` fields on call and field targets (including supported generic method targets), `ldstr` literal decoding with independent coverage, and pageable retained member summaries; SQLite snapshot inspection with table/column selection and bounded Evidence, requiring Node 24.x >=24.15 or Node 26+ with `DatabaseSync.limits`; packet capture inspection with bounded TShark output, compact capture summaries, retained pages, and discovery of installed protocol/field definitions, using Wireshark 4.6.6; recorded-signal decoding and discovery of installed decoders/options via caller-supplied `sigrok-cli` 0.7.2 (Linux-only, requires util-linux `prlimit`); QUIC capture inspection for retained aioquic 1.5.0 JSON qlog 0.3 objects with paging of traces, streams, and observations; Fiddler SAZ archive inspection with retained HTTP provenance and bounded decoding; WASM artifact inspection using bring-your-own WABT 1.0.42 without executing modules, plus import/export comparison across two core-module inspection Evidence records without requiring an active WABT installation; and PE resource inspection with read-only NativeAOT metadata parsing for x64 PE32+/.NET 8 RTR 9.1 inputs.

— Names exact new fields but no runnable command Concrete version requirement makes it directly adoptable Names tool version and output shape, no command shown Specific dependency and platform constraint named Names dependency versions but no entry point given Brief, no mechanism detail beyond one sentence Explains both static inspection and a diff mechanism Named platform/runtime target, thin on mechanism

`rea-agents-6.5.0` `rea-agents-6.4.0`

---

05 **WebMCP input schema field rename** BREAKING 75

JSON shape paths now contain typed property and array-element segments; `input_schema_shape` on WebMCP is replaced by `input_schema` and `input_schema_sha256`.

— Names exact replaced and replacing fields `rea-agents-6.3.0`

---

06 **New reverse-engineering tool integrations** NEW 73

Adds Frida instrumentation support with explicit local/remote target selection, caller-supplied scripts, retained messages, script-read/input diagnostics, and bounded message eviction; an optional native IDA MCP bridge alongside existing legacy/headless profiles, recorded against Linux IDA Free 9.4 and Windows IDA Pro 9.3; Android device inspection with split-APK acquisition, logs, screenshots, file transfers and explicit app lifecycle actions through ADB, plus manifest and localized-string decoding with Apktool; and JEB integration to browse project units and decompile selected items through a caller-started local JEB MCP server.

— Describes scope and controls but no example invocation Named IDA editions and platforms but setup not detailed Lists concrete capabilities and tooling but no exact commands Describes workflow but setup steps not given

`rea-agents-6.5.0` `rea-agents-6.4.0`

---

07 **Filesystem digest omission entry requirements** BREAKING

73

Filesystem digest omission entries now require `system_code`; budget and changed-file omissions use `null`, and unavailable-file omissions preserve the OS error code.

— Precise field-level schema requirement for consumers `rea-agents-6.3.0`

---

08 **Source-map identity and doctor skill state changes** BREAKING

73

Source-map URLs now live in target observations selected by `declared_url_sha256`; `doctor` skill identity no longer emits `installed_catalog_digest` — use the documented observation join and `identity.skill.state` instead.

— Gives exact replaced fields and the new lookup path `rea-agents-6.3.0`

---

09 **Objective-C attribute token schema simplification** BREAKING

71

Objective-C attribute tokens now contain only `name` and `value`; properties report nullable atomicity and readonly facts; class flags, ivar counts, and method requirement flags are nullable when unobserved.

— Specifies exact new token shape and nullability rules `rea-agents-6.3.0`

---

10 **Semantic graph evidence context requirement** BREAKING

68

Semantic graph facts now store evidence context IDs and locations; graph and semantic trace results require an inline `evidence_contexts` table.

— Names the new required table and its contents `rea-agents-6.3.0`

---

11 **Isolated JavaScript source analysis process** IMPROVED

63

Isolates JavaScript source analysis in an owned process with configurable heap and timeout limits, retaining completed facts and explicit unknowns when analysis runs out of resources.

– Describes mechanism and fallback behaviour clearly rea-agents-6.4.0

12 **Provider analysis output metadata requirements** BREAKING

60

Provider analysis outputs must explicitly supply metadata previously repaired with defaults; unsupported nullable facets must be reported as `null`.

– States new obligation but not which fields exactly rea-agents-6.3.0

THINNER COVERAGE BELOW

13 **New JavaScript application analysis capabilities** NEW

59

Adds RPC discovery for tRPC calls, HTTP-link URLs and named GraphQL operations in JavaScript applications and web bundles; large-analysis Evidence streaming to discover retained Evidence without loading every payload and inspect JavaScript graphs, binary layouts and function dossiers through summaries and selected views; and recording of Electron integrity mismatches in JavaScript application analysis results.

– Names protocols covered, no concrete entry point Explains behaviour change but no named interface States what is detected, no detail on how

rea-agents-6.4.0 rea-agents-6.3.0

14 **Removal of `truncated_scopes` statistic** DEPRECATED

47

The never-incremented `truncated_scopes` statistic is removed from JavaScript application analysis results and schemas.

– Names the removed field but little further context rea-agents-6.3.0

└──▷ **BREAKING ON UPGRADE**

! Historical captures without current producer accounting or transaction identity, and web diffs missing accessibility or storage dimensions, are now rejected — recapture with the current producer and update comparison consumers to include every dimension (see process-capture.md and browser-scenario-contract.md).

! JSON shape paths now contain typed property and array-element segments; `input_schema_shape` on WebMCP is replaced by `input_schema` and `input_schema_sha256`.

! Filesystem digest omission entries now require `system_code`; budget and changed-file omissions use `null`, and unavailable-file omissions preserve the OS error code.

! Provider analysis outputs must explicitly supply metadata previously repaired with defaults; unsupported nullable facets must be reported as `null`.

! Semantic graph facts now store evidence context IDs and locations; graph and semantic trace results require an inline `evidence_contexts` table.

! Source-map URLs now live in target observations selected by `declared_url_sha256`; `doctor` skill identity no longer emits `installed_catalog_digest` — use the documented observation join and `identity.skill.state` instead.

! The never-incremented `truncated_scopes` statistic is removed from JavaScript application analysis results and schemas.

! Objective-C attribute tokens now contain only `name` and `value`; properties report nullable atomicity and readonly facts; class flags, ivar counts, and method requirement flags are nullable when unobserved.

`rea-agents-6.8.0` NOTES ↗ **LATEST** STABLE

REA 6.8.0 adds agent environments from network captures and browser scenarios, evidence-gated write serving, state learning, row-oriented JSONL CLI exports, and Ghidra session improvements.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch rea-agents-6.8.0
https://github.com/morluto/rea.git
# already have the repo? check out this version:
$ git checkout rea-agents-6.8.0
```

Build local agent environments from network captures, browser scenarios, or a prompt, then serve derived reads and evidence-gated writes and compare with fresh application captures.

Project network captures and Evidence files into manifests, export AgentEnv bundles, and serve manifests as replay environments.

Render read filters, search, sorting, and paging from state; apply confirmed write effects with broader state targets and recording-based write admission.

Learn application-assigned values, view-specific clock spellings, write clocks from scenario timing, derived fields, and creates across sorted arrays and referenced entities.

Serve creates whose compatible read-backs add omitted fields or express timestamps at different precision, and whole-object updates regenerating observed slugs or modification times.

+4 more

└──▷ **BREAKING ON UPGRADE**

- ! `analyze_function` now defaults to a retained summary; set `detail: "complete"` for the previous inline dossier. CLI now requires `--evidence_output` when selecting a summary.
- ! `--format jsonl` now emits structured rows with a metadata control record; scalar selections and unsupported reports are refused. Use `--format json` for the previous whole-document output.
- ! `environment-serve` now serves derived state by default; use `--replay-only` for stateless replay. Read stop-summary `state` instead of `state_evidence_id`.
- ! MCP input advertisements default to compact; set `REA_MCP_INPUT_SCHEMA_PROFILE=full` for complete guidance.
- ! `instrument-with-frida` JSON returns canonical Evidence directly; read `normalized_result.cleanup_error` instead of the former top-level wrapper.
- ! Node.js runtime must satisfy `^22.23.4 || ^24.22.1 || >=26.12.1`; Node 23 and 25 are unsupported.
- ! `summary.by_evidence_authority` is keyed by Evidence authority; read former source-state counts from `summary.by_source_state`.
- ! Swift `MachoImportBinding` replaces the former Objective-C-specific import-binding type/schema.
- ! APK locale selectors such as `es-r419` replaced with `es-419` or `b+es+419`.
- ! XML plist unresolved scalars now use `$plist_type: "unknown"` wrapper with `element`, `literal`, and `reason` fields.

`rea-agents-6.7.0` NOTES ↗ STABLE

Adds retained JavaScript fact queries via `inspect_analysis_view`, multi-locale APK decoding with `--locales` / `locales`, `REA_JADX_OPERATION_TIMEOUT_MS` deadline control, and Electron `BrowserWindow` option resolution.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch rea-agents-6.7.0
https://github.com/morluto/rea.git
# already have the repo? check out this version:
$ git checkout rea-agents-6.7.0
```

Adds `inspect_analysis_view` to query exact application paths, selected properties, and filtered module references for retained JavaScript facts, with paging and complete row Evidence.

Adds `--locales` CLI flag and `locales` MCP parameter to request multiple APK resource locale tables in a single decode, each reporting its own status and coverage.

> Adds `REA_JADX_OPERATION_TIMEOUT_MS` environment variable to extend APK loading and upstream JADX operation deadlines together; Doctor reports the effective value and setup forwards an explicit selection.
>
> Adds resolution of stable local constant `BrowserWindow` options in Electron applications while retaining source provenance.

WAS THIS USEFUL? 👍 👎

whoop-mcp

Feedback ⓘ v1.1.0 NOTES ↗ RANK

Read-only WHOOP MCP server for recovery, sleep, HRV, strain, workouts, and personal health analytics in Claude, Codex, and GitHub Copilot.

> whoop-mcp's biggest addition this window is a suite of HYROX/triathlon performance MCP tools backed by a token-protected local dashboard and an append-only, claim-verifiable audit trail, alongside a redesigned chart-first dashboard, a new Today view, and broadened MCP client support.

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

01 **HYROX and triathlon performance MCP tools** NEW 85

v1.0.0 adds `get_hyrox_session_report`, `get_hyrox_performance`, `get_triathlon_performance`, `get_race_readiness`, `audit_performance_data`, and `verify_claim` for HYROX and triathlon performance tracking, `log_race_result` for recording validated HYROX station and triathlon leg splits locally, and `annotate_workout` to relabel, exclude, or restore a workout.

— Names every tool but not call parameters v1.0.0

02 **Token-protected local performance dashboard** NEW 75

`open_performance_dashboard` launches a token-protected local dashboard on `127.0.0.1` with Session, HYROX, Triathlon, Readiness, and Audit views; it shuts down automatically after 30 idle minutes.

— Names launch tool, address, and idle timeout v1.0.0

03 **Local race and audit data storage** NEW 75

Persistent race and audit data is stored locally in `~/.whoop-mcp/performance.json` and `~/.whoop-mcp/performance-audit.jsonl`, both created at file mode `0600`.

— Exact file paths and permission mode given v1.0.0

04 **Readiness band scoring with evidence** NEW 75

Computes a readiness band from recovery, training form (Edwards TRIMP CTL/ATL/TSB), specificity, sleep debt, and taper; each component carries its evidence and the band rule is returned verbatim.

— Explains mechanism and inputs, no direct call shown v1.0.0

05 **Dashboard Today view** NEW 75

Adds a Dashboard Today view using `get_today` to surface recovery, primary sleep, cycle strain, and the latest workout with source status and evidence.

— Names the backing tool and data surfaced v1.0.1

06 **Training view with MCP deep link** NEW 70

Adds a training view with an MCP deep link showing recorded WHOOP zone minutes by UTC week, workout strain/duration bubbles, and Edwards TRIMP fitness/fatigue/form calculations.

— Named metrics and calculation method, link format unspecified v1.0.2

07 **Metric audit trail and claim verification** NEW 65

Every metric reports its method, window, and records used or excluded; automatic suspect-record rules and an append-only audit log enable full audit trails, with `verify_claim` supporting verification against this history.

— Describes mechanism and names the verification tool v1.0.0

08 **Chart-first dashboard with theming** NEW 65

Adds a production chart-first dashboard with dark/light themes and locally bundled Archivo fonts (SIL OFL 1.1); install globally with `npm install -g whoop-ai-mcp` or run ad hoc via `npx whoop-ai-mcp`.

Install whoop-mcp globally to access the new chart-first dashboard and training views from the command line.

```
$ npm install -g whoop-ai-mcp
```

Run whoop-mcp without a global install to use the new dashboard and synthetic preview in a one-off session.

```
$ npx whoop-ai-mcp
```

— Exact install commands given, styling detail limited v1.0.2

09 **Workout and race result correction dialogs** NEW 60

Adds review dialogs for local workout corrections and race results, with Edit, Cancel, and Confirm save actions; pending saves cannot be submitted twice and failed saves preserve entries.

— Describes save-state mechanism but only a UI entry point v1.0.1

10 **Day-range presets and persistent dashboard filters** NEW

60

Adds completed-UTC-day presets for the last 7 or 28 days, persistent view filters, and refresh/retry controls, with load times explicitly separated from WHOOP sync times and cached reads supported.

— Specific presets named but navigation not detailed v1.0.1

THINNER COVERAGE BELOW

11 **35-day vitals timeline view** NEW

55

Adds a 35-day shared-timeline vitals view covering recovery, HRV, resting heart rate, and scored sleep stages, with historical mean ± SD bands.

— Describes metrics shown but no navigation path v1.0.2

12 **Session, zone, and readiness dashboard tiles** NEW

45

Adds a session recorded-segment ring, run-by-run HR build, WHOOP time-in-zone bar, five-part Readiness ring, training mix view, and Audit workout review tiles.

— Names six tiles but no usage mechanism v1.0.2

13 **Synthetic dashboard preview for screenshots** NEW

45

Adds a synthetic production-dashboard preview for repeatable browser checks and README/website screenshots without credentials or live WHOOP calls.

— Explains purpose, no exact command given v1.0.2

14 **Guidance-only setup for Claude web, ChatGPT, Grok** NEW

45

Adds guidance-only setup documentation for Claude web/mobile, ChatGPT, and Grok's xAI API, with explicit hosting, authentication, and compatibility limitations noted.

— Notes limitations but offers no working integration v1.1.0

15 **MCP client support for more editors and assistants** NEW

40

Adds MCP client support for Gemini CLI, Cursor, VS Code, OpenClaw, Perplexity, and Claude Desktop.

— Lists supported clients but no setup steps v1.1.0

WAS THIS USEFUL? 👍 👎

robloxstudio-mcp

Feedback ⓘ v3.4.0 NOTES ↗ RANK

MCP server for Roblox Studio runtime debugging, playtest control, screenshots/input, multiplayer testing, and per-peer server/client eval from AI agents.

> robloxstudio-mcp added a manage_experience tool for editing experience settings, atomic batched script edits with optimistic concurrency, async model generation, and tightened safety around search, writes, and multiplayer target routing.

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

01 **Batched atomic edits with concurrency control in edit_script_lines**

NEW 93

Adds an `edits` array to `edit_script_lines` that applies all changes as a single undo step, with `expected_revision` (obtained from `get_script_source`) for optimistic concurrency, `dry_run` for diff preview before committing, and `operation_id` for safe retries without duplicating the edit.

Apply a multi-line script patch atomically — preview the diff first, then commit only if the script hasn't changed since you read it.

```
$ # Step 1: read the current source and capture the revision
get_script_source({"path": "ServerScriptService/MyScript"})
# -> returns { "source": "...", "revision": 42 }

# Step 2: preview the diff
edit_script_lines({
  "path": "ServerScriptService/MyScript",
  "expected_revision": 42,
  "dry_run": true,
  "edits": [
    {"startLine": 10, "endLine": 10, "newText":
"print('hello')\n"}
  ]
})

# Step 3: commit (all edits land as one undo step)
edit_script_lines({
  "path": "ServerScriptService/MyScript",
  "expected_revision": 42,
  "operation_id": "patch-myScript-001",
  "edits": [
    {"startLine": 10, "endLine": 10, "newText":
"print('hello')\n"}
  ]
})
```

— Full mechanism plus a runnable multi-step example. v3.3.0

02 **Scoped, bounded search in search_objects** BREAKING 90

Adds `path`, `className` (IsA), and `maxResults` parameters to `search_objects`, capping results at 100 by default with a `truncated` flag. Query matching switched from Lua patterns to plain text, and an unknown `searchType` is now rejected with an error instead of silently returning no results. These are breaking changes: previously unbounded results now cap at 100, Lua-pattern queries no longer match as before, and unknown `searchType` values error instead of returning empty results.

Find all BasePart instances under a specific folder with a bounded result set, then check whether more matches were cut off.

```
$ search_objects({
  "path": "Workspace/Level1",
  "className": "BasePart",
  "maxResults": 50
})
# Check response: { "results": [...], "truncated": false }
```

— Every new parameter and breaking behavior named with example. v3.3.0

03 **Multiplayer target routing via role-suffixed instance IDs**

90

BREAKING

Role-suffixed `instance_id` values (e.g. `instance:…-client-2`) now correctly target that specific client or server in every tool that accepts `target`, and a conflicting `target` is rejected with `ambiguous_target`. `capture_screenshot`, simulated input, and `selection` with `action=view` now fall back to the playtest client when the edit DataModel is not connected, instead of failing with `target_role_required`. This is a breaking change: server-suffixed `instance_id` values now default to the server (not client-1) in `execute_luau`, `breakpoints`, `export_rbxm`, `import_rbxm`, `get_memory_breakdown`, and `get_scene_analysis`, so existing scripts relying on the old client-1 default must pass the controller `instance_id` explicitly.

Target a specific multiplayer client when running Luau in a Server Authority playtest, avoiding the old ambiguous client-1 default.

```
$ # Via MCP tool call — pass the role-suffixed instance_id to
route to client-2
execute_luau target=instance:abc123-client-2
script='print(game:GetService("Players"):GetPlayers())'
```

— Names affected tools, error codes, and migration note with example. v3.2.1

04 **Async job mode for generate_model** NEW

88

Adds `async=true` to `generate_model` to return a job ID immediately instead of blocking; callers use `get_request_status` with the new `wait_ms` parameter to poll or await the result, and `operation_id` to deduplicate retries.

Kick off a large model generation without blocking, then poll until it finishes — safe to retry without spawning a second job.

```
$ # Start async generation
generate_model({
  "prompt": "A medieval castle gatehouse",
  "async": true,
  "operation_id": "castle-gen-001"
})
# -> returns { "jobId": "job_abc123" }

# Wait up to 30 s for the result
get_request_status({
  "jobId": "job_abc123",
  "wait_ms": 30000
})
```

— Exact parameters and a runnable polling example. v3.3.0

05 **Safer find-and-replace in scripts** IMPROVED 73

`find_and_replace_in_scripts` now shows a preview step of exactly what will change per script before applying, and rolls back already-changed scripts atomically if a write fails partway through. It also refuses to apply a partial replacement when the total would exceed `maxReplacements`, instead reporting the total count, and adds `classFilter: "Script"` scoping so edits no longer bleed into LocalScripts and ModuleScripts when only Script is specified.

— Names every flag and behavior but no runnable example. v3.4.0

06 **Ambiguous path write protection** BREAKING 73

Refuses writes through ambiguous paths (where sibling instances share a name) in `set_properties`, `insert_asset`, `import_rbxm`, and `find_and_replace_in_scripts` (without `dryRun`); reads now report `ambiguousPaths` and `pathWarning` instead of silently mutating or matching an arbitrary sibling.

— Names affected tools and fields but lacks an example. v3.3.0

07 **manage_experience tool for experience settings** NEW 70

New `manage_experience` tool reads and updates the connected experience's `title`, `description`, server size, voice chat, supported devices, private server price, and social links, letting agents configure experience settings without opening the Studio settings panel.

Read and update experience metadata — title, server size, voice chat — directly from your AI tool without opening the Studio settings panel.

```
$ # Via Claude Code after adding the MCP server:
# Prompt your AI: 'Use manage_experience to set the title to "My
Game", server size to 20, and enable voice chat.'

# Add the server first:
claude mcp add robloxstudio -- npx -y @chrrxs/robloxstudio-
mcp@latest --auto-install-plugin
```

— Names tool and fields with an example, but no return schema. v3.4.0

THINNER COVERAGE BELOW

08 **Mouse move and scroll simulation** NEW 52

Adds `move` and `scroll` (with `wheel`) actions to `simulate_mouse_input`; `move` is rejected when touch device emulation is active.

— Names actions and a constraint but no usage example. v3.3.0

└──▷ BREAKING ON UPGRADE

- ! `search_objects` now caps results at 100 by default (previously unlimited); callers relying on unbounded results must check the new `truncated` field.
- ! `search_objects` now matches queries as plain text instead of Lua patterns; existing queries that relied on Lua pattern syntax will no longer match as expected.
- ! `search_objects` now rejects an unknown `searchType` with an error instead of silently returning no results.
- ! Writes through ambiguous paths (`set_properties`, script edits, `insert_asset`, `import_rbxm`, `find_and_replace_in_scripts` without `dryRun`) are now refused when siblings share a name, instead of silently mutating an arbitrary match.
- ! Server-suffixed `instance_id` values now default to the server (not client-1) in `execute_luau`, `breakpoints`, `export_rbxm`, `import_rbxm`, `get_memory_breakdown`, and `get_scene_analysis` — existing scripts that relied on the old client-1 default must pass the controller `instance_id` explicitly.

WAS THIS USEFUL? 👍 👎

rails-ai-context

Feedback ⓘ v5.37.0 NOTES ↗ RANK

35 MCP tools providing Rails app ground truth to coding agents: schema, models, routes, controllers, views, jobs, and conventions. Works when the app cannot boot.

> rails-ai-context's eight releases this window center on a sweeping secret-redaction hardening pass across rails_query, rails_read_logs, rails_get_view, rails_get_config, rails_search_code and rails_review_changes, alongside a new Claude Code plugin install path, new uninstall/mcp-entry CLI commands, automatic update checks, and performance work on rails_validate and doctor.

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

01 **Claude Code plugin install and diagnostics** NEW 93

Installs the MCP server as a Claude Code plugin via `claude plugin install rails-ai-context --marketplace crisnahine/rails-ai-context --scope project`, writing no context files or MCP config and surviving gem updates without changing `allowedMcpServers`. `doctor` gains plugin-aware diagnostics, reading Claude Code's settings, install registry, `/mcp` toggles and MCP policy per session folder, naming each blocking rule and the file that holds it, and warning about duplicate servers with the commands to remove them.

Add rails-ai-context as a Claude Code project plugin so every session in the app gets MCP tools automatically, without touching .mcp.json or context files.

```
$ gem install rails-ai-context
cd /path/to/your-rails-app
claude plugin install rails-ai-context --marketplace
crisnahine/rails-ai-context --scope project
```

— Exact install command plus diagnostic mechanism named v5.36.9

02 **Secret filtering extended for plural, list and bracketed names**

IMPROVED 90

`rails_read_logs` and `rails_query` now filter secret values under plural names (`api_keys` , `tokens` , `secrets` , `passwords` , `accessTokens`), list-valued secrets (`recovery_codes` , `backup_codes` , `recoveryCodes` , `backupCodes`), bracketed query parameters with a secret word anywhere in the name (e.g. `user[password][new]=`), and comma-separated env values like `API_KEYS=a,b` , while counts (`tokens_used` , `token_count` , `prompt_tokens`) and structural keys (`keys` , `primary_keys` , `cache_keys`) remain visible. `rails_query` still refuses a column

whose name contains `token` unless `query_allowed_columns` lists it explicitly, configurable via `config.rails_ai_context.query_allowed_columns`.

Expose a token-related query column that would otherwise be blocked by the secret-name filter.

```ruby
Rails.application.configure do
  config.rails_ai_context.query_allowed_columns =
["token_version"]
end
```

— Names exact filtered patterns and the config key to unblock a column v5.36.9

---

03 **Unattended install flags and copy-paste prompts** BREAKING

90

Adds `--tools claude,cursor` (or `all`), `--mcp`, `--cli`, `--prune`, `--defaults` (init only), and `--hook` (generator only) flags to `init` and the install generator for fully unattended installs. Adds `docs/PROMPTS.md` with copy-paste prompts for every install scenario — Gemfile and standalone installs, CLI-only, MCP-only, team, folder-of-apps, private, Docker, HTTP, and MCP clients the gem writes no config for — compatible with Claude Code, Cursor, GitHub Copilot, OpenCode, and Codex CLI. A re-run with `--defaults` now preserves the app's recorded AI tools and setup instead of resetting to all five tools and the MCP setup, changing the outcome of unattended re-runs that relied on the old default.

Run a fully unattended install for Claude Code and Cursor in CI or from an AI tool's terminal without any interactive prompts.

```
$ bundle exec rails generate rails_ai_context:install --tools
claude,cursor --defaults
```

— Lists every flag verbatim plus a runnable install command v5.35.0

---

04 **Redaction engine recognizes more secret patterns** IMPROVED

88

Redaction now filters secrets passed as CLI flags with `NAME=value` syntax (`--password=...`, `--requirepass`), dashed HTTP header names like `X-Api-Key` and `x-auth-token`, `Authorization` header credentials (`Bearer`, `Basic`, `Token`) when 16+ characters, environment variables read by Postgres (`PGPASSWORD`), MySQL (`MYSQL_PWD`), and redis-cli (`REDISCLI_AUTH`), camel-cased secret names (`accessToken`, `jwtSecretKey`, `clientSecret`), bare flag values like `redis-server --requirepass ...`, and secrets named on a separate YAML or JSON line including

Heroku `app.json` `value:` fields, `render.yaml` / Kubernetes `key:` / `value:` sibling pairs, and `config/sidekiq.yml` symbol keys like `:password: ...` — while preserving Stimulus data attributes such as `data: { password_visibility_target: 'input' }`, switches (`--no-password`), paths (`--private-key ~/.ssh/id_ed25519`), and BuildKit `--secret id=...,env=...` syntax.

— Enumerates every new pattern and every preserved exception verbatim v5.33.0

05 **Automatic update checks with opt-out** NEW 86

The standalone binary now checks rubygems.org for newer releases at most once a day (in a detached child process) and prints a one-line upgrade notice on stderr in terminal sessions; the MCP server surfaces the notice in its instructions and at the end of `rails_onboard`. Set `RAILS_AI_CONTEXT_NO_UPDATE_CHECK=1` to disable it; state is cached under `$XDG_STATE_HOME` (`~/.local/state/rails-ai-context/latest`).

Suppress the daily rubygems.org version check in CI or scripted environments where the stderr notice would be noise.

```
$ RAILS_AI_CONTEXT_NO_UPDATE_CHECK=1 rails-ai-context tool onboard
```

— Exact env var and cache path with a runnable suppression command v5.36.4

06 **uninstall command** NEW 86

Adds `rails-ai-context uninstall` command to cleanly remove per-tool context files, MCP entries, `.rails-ai-context.yml`, `.ai-context.json`, the initializer, `.gitignore` lines, and the pre-commit hook while preserving user-written content.

Cleanly remove rails-ai-context from an app while keeping user-edited sections in shared files like CLAUDE.md and existing MCP servers.

```
$ bundle exec rails-ai-context uninstall
```

— Runnable command naming every file and entry it removes v5.35.0

07 **Path and pattern handling hardened against traversal and resource exhaustion**

IMPROVED 85

Sensitive-pattern matching now follows `.gitignore` semantics: a `/` in a pattern matches below the app root, a leading `/` anchors at the root, a trailing `/` covers a folder and its contents, and `**/` matches any depth, fixing gaps where engine, pack, and test/dummy copies of `config/database.yml` were readable. `rails_search_code` cleans paths before searching so `.` segments and symlinks can

no longer bypass sensitive-pattern matching, and with `match_type:'trace'` now validates its `path` argument against the app root, preventing traversal via `path: '../../elsewhere'`. Directory symlinks pointing outside the app's git work tree are no longer followed during walks or packs. `rails_get_view` and `rails_get_partial_interface` now treat the supplied path as plain text instead of a glob, preventing crafted `{a,b}` patterns from holding a server thread for minutes.

— Enumerates every tool and pattern-matching rule fixed `v5.36.8` `v5.33.0`

---

08 **rails_query hardened against secret-column bypass** BREAKING

82

`rails_query` now refuses a `PROCEDURE` clause on MySQL and MariaDB, blocking SELECT ... PROCEDURE ANALYSE() from returning secret-column metadata under unredacted column names. It also now refuses queries that reference sensitive columns through aliases, expressions, CTEs, subqueries, or set operations, and refuses all UNION variants (`UNION DISTINCT SELECT`, parenthesised UNION (SELECT ...)) in addition to `UNION SELECT` and `UNION ALL SELECT` — queries using these forms that previously returned results will now be blocked.

— Names the exact SQL constructs newly blocked `v5.36.8` `v5.33.0`

---

09 **Secret filtering in rails_get_view full controller listings**

IMPROVED

81

`rails_get_view` now filters secrets in a controller's full listing (`controller:'posts', detail:'full'`), including inline assignments like `<% api_key = '...' %>` and `render` call locals such as `token: '...'`, and covers mailer views (`controller:'user_mailer'`).

— Gives exact tool parameters to reproduce `v5.33.0`

---

10 **rails_review_changes validates refs and redacts diffs** BREAKING

80

`rails_review_changes` now validates its `ref` argument to a commit SHA before passing it to `git diff` / `git log`, refusing refs that start with a dash and preventing `ref: '--output=<path>'` from writing files via git; callers passing non-commit refs now get an error instead of a diff. It also redacts secrets in diff lines (added and removed) and shows only line counts for files matching `sensitive_patterns` such as `config/database.yml` and `.env`.

— Names exact validation rule and the breaking behavior change `v5.33.0`

---

11 **mcp-entry command for unsupported clients** NEW

80

Adds `rails-ai-context mcp-entry` command to print a ready-to-paste MCP server entry (with full Ruby binary, executable, app paths, and gem environment) for

unsupported clients such as Claude Desktop, Windsurf, Zed, and JetBrains IDEs.

Get a ready-to-paste MCP server entry for a client the gem does not configure automatically, such as Windsurf or a JetBrains IDE.

```
$ bundle exec rails-ai-context mcp-entry
```

— Runnable command with named target clients `v5.35.0`

---

12 **Partial-scan mode for rails_validate** IMPROVED 78

Adds partial-scan mode to `rails_validate` with `level: "rails"`: Brakeman (8.1.0) now parses only the checked files and the app's models instead of the whole app, changing a single-file scan from ~0.02s to ~0.32s at 250 models and from ~0.04s to ~0.92s at 700 models.

— Names parameter and exact benchmarks but no full command shown `v5.37.0`

---

13 **Line-based lookup in rails_get_edit_context** NEW 75

Adds a `line` parameter to `rails_get_edit_context`, showing `context_lines` either side of that line so a stack-trace line number needs no search text; `near` is only required without it.

— Names exact new parameter and its interaction with existing one `v5.36.5`

---

14 **Redaction extended to gems, Procfile and edit-context labels** IMPROVED 73

The gems section and `rails://gems` resource now redact credentials embedded in Gemfile `git:` URLs (`https://user:token@host/...`), replacing userinfo and token-format values with `[FILTERED]`. Procfile commands written to `.ai-context.json` by the DevOps section are now redacted, filtering secrets like `redis-server --requirepass ...` or `DB_PASSWORD=... bundle exec sidekiq` before they reach the committed context file. `rails_get_edit_context` now builds the enclosing-method label from filtered lines and names endless methods without their body, preventing credentials in `def headers = { 'Authorization' => 'Bearer ...' }` from appearing in the label; `near` is also matched against the filtered lines.

— Names every surface and example credential pattern redacted `v5.33.0`

---

15 **SQL statement secret filtering in rails_read_logs** IMPROVED 71

`rails_read_logs` now filters secret-column values inline in MySQL/MariaDB statements and `insert_all` / `upsert_all` output, covering INSERT, UPDATE, WHERE, CAST(), LOWER(), `IN`, `BETWEEN`, JSON paths, and database error messages.

— Names tool and exact SQL constructs covered `v5.36.8`

16 **DSN and webhook redaction in env config** IMPROVED 71

rails_get_config(section: 'env') now redacts DSN and webhook URLs in Dockerfile `ENV` / `ARG` values, `.env.example` entries, and code defaults, catching Sentry DSNs, Slack webhooks, query-string tokens, `BUNDLE_<HOST>` credentials, and key-like strings that the generic URL-password filter missed.

— Names exact call and credential types newly caught v5.36.8

17 **Model and controller tools surface more metadata** IMPROVED 71

`rails_get_model_details` now adds an `[UNAVAILABLE]` annotation naming any `*::Patches::<Model>Patch` modules the static tier could not read, so callers know when plugin-added behaviour is missing, and lists each `has_attached_file` attachment under a Macros section. The controller listing now marks `permit!` and surfaces strong-params methods defined on parent controllers, naming the parent.

— Names exact annotation, module pattern and section labels v5.34.0

18 **TestHelper raises on tool errors instead of silent pass** 70

BREAKING

`TestHelper#execute_tool` now raises when the tool returns an error, and `execute_tool_with_error` now fails the test when the call succeeds, so a test against a broken tool no longer passes silently. A suite that read an error text through `execute_tool` must switch to `execute_tool_with_error`.

— Clear migration path named, no code example given v5.37.0

19 **rails_security_scan shows verbatim source statement** IMPROVED 70

`rails_security_scan` at `detail:'full'` now shows the warning's source statement as written in the file — up to 12 lines, filtered — instead of Brakeman's variable-inlined version, which stripped names needed for redaction.

— Names exact parameter and line limit v5.33.0

20 **Static tier parses config without booting the app** IMPROVED 68

The static tier now parses `sensitive_patterns`, `excluded_paths`, and `skip_tools` values from `RailsAiContext.configure` blocks in Ruby source files without booting the app, so settings are never silently dropped under `--no-boot` or a failed boot.

— Names exact config keys, no example invocation given v5.36.8

21 **Redaction engine now runs in linear time** IMPROVED 65

Redaction against long tokens now runs in linear time — a 16 KB hex string that previously took 1.5–2 seconds on Ruby 3.1/3.2 (and a 64 KB string over 20 seconds) now completes in a fraction of a second, eliminating the Rails 8 regex timeout on 1 MB strings under Ruby 3.3+.

— Gives exact before/after benchmarks, no direct invocation v5.33.0

22 **Caching speeds up env and doctor introspection** IMPROVED

60

The env section now caches each source file's scan until the files change, making repeated calls cost milliseconds instead of about a second, and the read runs without holding the shared cache's lock so other tools do not wait on it. `doctor` now introspects once, sharing its run between the health check and the context files' dry run, cutting doctor time by about half on large apps and fixing false health-check failures for introspectors that fail only when run alone.

— Explains mechanism and before/after timing, no direct invocation v5.37.0

THINNER COVERAGE BELOW

23 **auto_mount endpoint requires explicit opt-in** BREAKING

50

The mounted engine and `auto_mount` endpoint now return HTTP 403 in all environments except development and test; production and staging deployments that relied on the endpoint answering without opt-in will break until the app opts in using the named setting.

— Breaking change named but the opt-in setting itself isn't spelled out v5.33.0

24 **Symlink safety for install and uninstall** IMPROVED 45

Install and uninstall no longer write through or delete files carried by a symlinked directory; such files are left in place and named as left behind a symlink the cleanup does not follow.

— Describes behavior with no command or flag to invoke v5.37.0

└──▷ BREAKING ON UPGRADE

! A test suite that read an error text through `execute_tool` must call `execute_tool_with_error` instead — `execute_tool` now raises on a tool error rather than returning it.

! Running the install again without flags previously defaulted to all five AI tools and the MCP setup; it now defaults to whatever the app already has recorded, which changes the outcome of unattended re-runs that relied on the old default.

! The mounted engine and `auto_mount` endpoint now return HTTP 403 in all environments except development and test; production and staging deployments that relied on the endpoint answering without opt-in will break until the app opts in using the named setting.

! `rails_query` now refuses queries that reference sensitive columns through aliases, expressions, CTEs, subqueries, or set operations — queries that previously returned results will be blocked.

! `rails_query` now refuses all UNION variants (`UNION DISTINCT SELECT` , parenthesised UNION (SELECT ...)) in addition to `UNION SELECT` and `UNION ALL SELECT` — queries using those forms will be blocked.

! `rails_review_changes` now refuses `ref` values that do not name a commit or that start with a dash — callers passing non-commit refs will receive an error instead of a diff.

WAS THIS USEFUL?  

opentakeoff

Feedback ⓘ mcp-v0.9.99 NOTES ↗ RANK

Open-source PDF takeoff for construction and flooring, designed for AI agents to drive via MCP. Measures rooms, symbols, and materials with full provenance in-browser.

> opentakeoff added a full estimate workflow — a pricing engine, Excel export, and watermarked bid proposal PDFs — alongside new MCP estimate verbs exposed over the canvas's estimate engine, on-device OCR for text embedded in vector sheets, a guided first-takeoff onboarding flow, and several schedule-import robustness improvements.

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

01 Schedule import robustness: headers, legends, plan-presence, recovery

IMPROVED 70

Extends schedule import to handle alias headers, a code column printed second, and headerless legends; adds per-code plan-presence reporting indicating whether each imported code appears on the plans; and requires a scanned-code review step while recovering omitted numeric rows during import.

– Covers several concrete import scenarios but no exact UI/command reference. mcp-v0.9.98

02 Estimate workflow: company setup, panel, bid PDF NEW

65

Adds a Company setup screen covering identity, estimating defaults, and proposal text; an Estimate panel with Excel export for takeoff estimates; and Bid proposal PDF generation, with an OpenTakeoff watermark applied to every exported PDF.

– Enumerates screens and export formats but no exact navigation steps. mcp-v0.9.99

THINNER COVERAGE BELOW

03 Estimate pricing engine with tax and overhead NEW

55

Adds an estimate pricing engine supporting unit prices, tax, overhead, and profit calculations.

– Names calculation components but no mechanism or UI path given. mcp-v0.9.99

04 On-device OCR for embedded images in vector sheets NEW

55

Reads and copies text from pictures embedded in vector sheets via on-device OCR.

– Mechanism named (on-device OCR) but no command or UI path. mcp-v0.9.98

05 **MCP estimate verbs over the canvas engine** NEW 45

Adds MCP estimate verbs over the canvas's estimate engine, exposing estimate operations via the Model Context Protocol in this release.

— Names the MCP surface but not individual verb names or schema. mcp-v0.9.99

06 **Sheets view grouped by title-block sheet numbers** IMPROVED 40

Groups PDFs in a tree organised by title-block sheet numbers in the Sheets view.

— Single-line description with no deeper mechanism. mcp-v0.9.99

07 **Guided first-takeoff onboarding and AI connection setup** NEW 40

Adds a guided first-takeoff flow and a Connect your AI setup to onboard new users.

— Thin onboarding description without workflow detail. mcp-v0.9.98

WAS THIS USEFUL? 👍 👎

godot-ai

Feedback ⓘ v4.3.1 NOTES ↗ RANK

Production-grade MCP server and AI tools for the Godot engine. A Snap to install.

> godot-ai v4.3.1 hardens write-path security against Windows alias spellings and improves mouse input simulation by carrying held button state on motion events.

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

01 **Held mouse buttons on input_mouse motion** IMPROVED 52

`input_mouse` motion events now carry held mouse button state, enabling accurate mouse-state replay in game input.

— Names exact event type but no usage example given v4.3.1

02 **Windows alias path-traversal hardening** IMPROVED 33

Write paths now refuse Windows alias spellings, hardening the tool's defenses against path-traversal attacks.

— Names the risk but no concrete endpoint or config detail v4.3.1

WAS THIS USEFUL? 👍 👎

godot-mcp

Feedback ⓘ v0.33.0 NOTES ↗ RANK

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

> Godot-MCP's biggest change this window is RQL (Runtime Query Language), which replaces the prior editor data tools and absorbs extensions, shader/layer/mesh types, and project data into one query surface — a breaking migration. Alongside it, the tool gained two separate script-execution paths (in-game via Roslyn, headless via Code-Exec), standalone CLI/App guides, a new Tilemap NuGet extension, and smaller additions to node editing, log streaming, plugin installation, and MCP server configuration.

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

01 RQL replaces editor data tools and expands query surface

85

BREAKING

Introduces RQL (Runtime Query Language) as the replacement for the editor data tools via `McpPlugin` 8.8.1 and `Rql` 0.1.1, giving a single query language for the whole Godot editor, and converts Godot extensions into RQL modules. `McpPlugin` 8.9.0 adds a shared loopback token rule, host-bound credentials, and Godot error hints for the connection layer. RQL is then expanded with shader parameters, layer names, `MeshLibrary`, and `Projection` / `RID` / `Callable` type adapters, plus new RQL models for groups, signals, project settings, input map, autoloads, `TileSet`, and `Theme`. Breaking: any workflow depending on the previous editor data tools is broken and existing extensions must be repackaged as RQL modules or they will not load.

— Full mechanism, version numbers, and breaking migration path named v0.31.0 v0.30.0 v0.29.0

02 Headless script execution via Code-Exec IMPROVED

75

Adds `script-execute` capability on Code-Exec and `godot-cli exec` headless sessions (Code-Exec phase 4), enabling AI agents to run scripts without the editor open; later upgraded to use Code-Exec 0.2.1, dropping previous host/Console workarounds.

— Named command, headless scenario, and version upgrade with before/after v0.31.0 v0.30.0

03 node-get-data command for node property reads NEW

60

Adds `node-get-data` command that reads a node's stored properties, providing the correct input for `node-modify` round-trips.

— Named command pair is directly usable v0.28.0

04 **Incremental log retrieval via sinceSequence cursor** NEW

60

Adds `sinceSequence` cursor parameter to `console-get-logs`, enabling incremental log retrieval from a known sequence position.

— Named parameter and command make this directly usable v0.25.2

THINNER COVERAGE BELOW

05 **In-game script execution via Roslyn** NEW

50

Adds opt-in in-game script execution via Roslyn, enabling C# scripts to be compiled and run inside a live running Godot game at runtime.

— Explains mechanism but no concrete flag or command given v0.32.0

06 **install-plugin pins NuGet versions** IMPROVED

50

The `install-plugin` command now pins the NuGet versions of the addon it installs, ensuring reproducible dependency resolution across environments.

— Names command and behaviour change, modest mechanism detail v0.26.0

07 **Standalone CLI and App guides with host-aware commands** NEW

45

Adds standalone CLI and App guides with global metadata to the skills system, and generates executable commands targeting the host platform for cross-platform correctness.

— Describes the addition but no concrete command syntax shown v0.33.0

08 **Tilemap NuGet extension package** NEW

45

Ships `com.IvanMurzak.Godot.MCP.Tilemap` as a new source-only NuGet package, independently versioned from core and other extensions.

— Names the package but no usage detail beyond versioning tilemap-v1.0.0

09 **Tool-timeout support for local MCP server** NEW

40

Upgrades `McpPlugin` to 8.11.0 and adds `tool-timeout` support when starting the local MCP server.

— Named flag and version but no usage context given v0.31.0

10 **Exported games exclude dev-only assemblies** IMPROVED

40

Excludes RQL, Code-Exec, and Roslyn assemblies from exported games, keeping production builds clean.

— Names excluded assemblies but limited further detail v0.31.0

11 **CLI editor sessions via gamedev-cli-core** IMPROVED 30

Upgrades the CLI to adopt `@baizor/gamedev-cli-core` 0.7.1 editor sessions.

— Bare dependency bump with no stated behaviour change v0.31.0

└──▷ BREAKING ON UPGRADE

! RQL replaces the editor data tools — any workflow depending on the previous editor data tools is broken; upgrade requires migrating to RQL (McpPlugin 8.8.1, Rql 0.1.1).

! Godot extensions are now RQL modules — existing extensions must be repackaged as RQL modules or they will not load.

WAS THIS USEFUL? 👍 👎

thunderbird-mcp

Feedback ⓘ v0.10.0 NOTES ↗ RANK

MCP server for Thunderbird - enables AI assistants to access email, contacts, and calendars

> thunderbird-mcp shipped nine new MCP tools this window — including scheduled draft delivery, a dry-run compose preview, attachment search, and saved-search virtual folders — alongside calendar recurrence/attendee support and a wave of security hardening covering encrypted-message access, attachment paths, connection-file trust, and mail filter behavior.

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

01 Calendar event reminders, recurrence, and attendees NEW

89

Extends `createEvent`, `updateEvent`, and `listEvents` to handle display reminders via a `reminders` array (e.g. `reminders: [1440, 0]`) at or before event start, RRULE recurrence support, a `recurrenceId` parameter to update or delete a single occurrence of a recurring event, and attendee management; `listEvents` output now includes organizer, attendees, and the user's response status. Meetings (events with attendees) are read-only through MCP by default while `Block skipReview` is enabled — changing, deleting, or adding attendees is blocked until the setting is turned off. `listCalendars` also reports a `disabled` field, and targeting a disabled calendar now returns an error instead of an empty list.

Update only a single occurrence of a recurring event without affecting the rest of the series.

```
$ {"tool": "updateEvent", "eventId": "<event-id>",
"recurrenceId": "20250610T090000Z", "title": "Team Sync
(rescheduled)", "start": "2025-06-10T10:00:00", "end": "2025-06-
10T11:00:00"}
```

— Multiple named parameters, fields, and a runnable example. v0.9.0 v0.10.0

02 Bridge connection-file discovery and hardening IMPROVED

88

Adds `THUNDERBIRD_MCP_CONNECTION_FILE` environment variable to manually specify the connection file path when automatic discovery fails in sandboxed environments such as Firejail; discovery is restricted to files whose Thunderbird process is still running, belongs to the current user, and (on Linux/macOS) resides in a folder private to that user, or on Windows inside the user's own temp folder. The bridge now also verifies `connection.json` as a regular file owned by the current user with owner-only

permissions before use, refusing symlinks or looser-permission files, including when pointed to via `THUNDERBIRD_MCP_CONNECTION_FILE`.

Override connection-file discovery when running Thunderbird inside a Firejail sandbox where process-ID isolation breaks automatic detection.

```
$ THUNDERBIRD_MCP_CONNECTION_FILE=/run/user/1000/thunderbird-mcp/connection.json node mcp-bridge.cjs
```

— Named env var, exact runnable command, and security checks. v0.8.0 v0.9.1

---

03 **Draft scheduling, threading, and signature controls** NEW

86

Adds `sendAt` to `saveDraft` to schedule delivery via the Send Later add-on (requiring `sendMail` to be enabled, with `skipReview` governing scheduled drafts), a `saveAsDraft` option on `replyToMessage` to save a threaded reply directly to Drafts, and `inReplyTo` / `references`, `replaceMessageId`, and `includeSignature` parameters on `saveDraft` (plus `includeSignature` on `sendMail` with `skipReview`) for threading, overwriting existing drafts, and controlling signature insertion.

Schedule a draft to send in 30 minutes without opening a compose window — useful for timed follow-ups from an automated workflow.

```
$ { "tool": "saveDraft", "arguments": { "to": "colleague@example.com", "subject": "Follow-up", "body": "Hi, just following up.", "sendAt": "2025-09-01T10:30:00Z" } }
```

Reply to a message and save it as a draft for review before sending, preserving the thread.

```
$ {"tool": "replyToMessage", "messageId": "<msg-id>", "body": "Thanks for the update — will follow up tomorrow.", "saveAsDraft": true}
```

— Named parameter, dependency, and runnable example call. Named parameters with a runnable threading example.

v0.10.0 v0.9.0

---

04 **New folder, header, attachment, and virtual-folder tools** NEW

86

Adds `refreshFolder` and `refreshCalendar` tools to sync an IMAP, Exchange, or Owl folder or calendar (waiting up to 25 seconds for the result), `getMessageHeaders` and

`batchGetMessageHeaders` tools to read headers by Message-ID (up to 200 per call, including threading references) without parsing message bodies, `searchAttachments` to find messages by attachment filename or MIME type using local copies up to 16 MiB and reporting skipped messages, and `createVirtualFolder`, `listVirtualFolders`, and `deleteVirtualFolder` tools for managing saved searches.

Find all messages with a PDF attachment in the Inbox to audit file delivery without downloading or decrypting anything.

```
$ { "tool": "searchAttachments", "arguments": { "folder": "Inbox", "mimeType": "application/pdf" } }
```

— Named tools and timeout, no usage example provided. Named tools and limit, lacks a usage example. Named tool, limits, skip reasons, and runnable example. Tool names given but no parameters or example shown.

v0.10.0

---

05 **Encrypted message access controls** BREAKING 80

When 'Allow MCP clients to read encrypted messages' is off (the default), `getMessage` returns a placeholder with `encryptedContentWithheld: true` for OpenPGP and S/MIME messages, `searchMessages` and `getRecentMessages` show outer subjects without previews for previously-decrypted OpenPGP messages, and `displayMessage` refuses to display encrypted messages.

Enable encrypted-message access and forward/reply filter actions for a trusted local MCP client via the Options page.

> 📍 In the Thunderbird Add-ons Manager, open the thunderbird-mcp Options page, then enable 'Allow MCP clients to read encrypted messages' and Allow automatic Forward/Reply filter actions.

— Named field, affected tools, and an enabling-steps example. v0.8.0 v0.9.1

---

06 **Mail filter and folder-action restrictions** BREAKING 80

Blocks use of the Outbox (Unsent Messages) as a destination in `updateMessage` (`moveTo`, `copyTo`), `createFolder`, `moveFolder`, and `renameFolder`; filters using 'is / isn't in address book' conditions are skipped by `applyFilters` when mail accounts are restricted and Allow all address books is off; adds Allow automatic Forward/Reply filter actions option (Options page, off by default) without which MCP clients cannot create, change, or run filters that forward or reply; `applyFilters` now runs only enabled rules marked for manual runs; and filter date conditions now accept only the `YYYY-MM-DD` format.

Enable encrypted-message access and forward/reply filter actions for a trusted local MCP client via the Options page.

> 📍 In the Thunderbird Add-ons Manager, open the thunderbird-mcp Options page, then enable 'Allow MCP clients to read encrypted messages' and `Allow automatic Forward/Reply filter actions`.

— Every endpoint, option, and format named across releases. `v0.8.0` `v0.9.0` `v0.9.1`

---

07 **dryRunCompose preview tool** NEW 78

Adds `dryRunCompose` tool to preview what `sendMail` would do — identity, recipients, automatic Cc/Bcc, attachments, estimated size — without sending, saving, or opening anything.

Verify a message composition will succeed before committing — catch missing recipients or oversized attachments in CI or automation.

```
$ { "tool": "dryRunCompose", "arguments": { "to":
"recipient@example.com", "subject": "Report", "attachments":
["/reports/monthly.pdf"] } }
```

— Named tool, scope, and example call given. `v0.10.0`

---

08 **Attachment path security restrictions** BREAKING 73

Attachment-by-path now blocks dot-directories (`~/.ssh` , `~/.config` , `.env`), `AppData` , the macOS home `Library` , credential-like file names, UNC/network paths, and Windows alternate data streams, and a refused attachment fails the entire send, draft, or reply operation including `saveDraft` . The bridge separately refuses attachment files with more than one hard link, and refuses attachments on Windows mapped network drives by resolving them to network paths.

— Exhaustive list of blocked path types, no example call. `v0.8.0` `v0.9.1`

09 **copyTo action for message duplication** NEW 71

Adds `copyTo` action on `updateMessage` to copy a message to another folder without moving it, enabling use cases like adding Gmail labels.

Copy a message to a Gmail label folder without removing it from its current location.

```
$ {"tool": "updateMessage", "messageId": "<msg-id>", "copyTo": "mailbox://user@gmail.com/[Gmail]/Important"}
```

— Named action with a concrete runnable example. v0.9.0

10 **Message body sanitization and Markdown conversion limits** IMPROVED 71

Message body output now drops hidden HTML, invisible characters, and bidi control characters; Markdown output keeps only `http`, `https`, and `mailto` links, converts images to alt text, and truncates HTML over 2 MiB with a note.

— Concrete sanitization rules and size limit, no example. v0.8.0

11 **fullRecipients parameter for complete recipient lists** IMPROVED 67

Adds `fullRecipients: true` parameter to `searchMessages` and `getRecentMessages` to retrieve complete recipient lists; by default these tools cap output at 3 recipients and 3 Cc addresses plus (+N more).

— Named parameter and default cap, no example call. v0.10.0

12 **Contact write tools require explicit opt-in** BREAKING 60

`createContact`, `updateContact`, and `deleteContact` are disabled by default after upgrading; re-enable them by turning on Allow contact writes in Options.

— Clear migration step via a named Options toggle. v0.10.0

THINNER COVERAGE BELOW

13 **skipReview safety restrictions for sending** BREAKING 59

`skipReview: true` is now refused when there are no recipients, or when the message would exceed `mailnews.message_warning_size` (20 MiB by default).

— Exact threshold named, no remediation path described. v0.10.0

14 **Accurate delivery reporting for skipReview sends** IMPROVED

59

Direct sends (`sendMail` with `skipReview`) now report success only after delivery fully completes; previously, rejected recipients or an unreachable server could be reported as sent.

— Before/after behavior described, no example given. v0.9.0

---

15 **Options page access control and status indicators** NEW

58

Adds Allow all calendars and Allow all address books options (Options page, off by default) to unblock calendar and address book access when mail account restrictions are in force, and adds an Options page indicator showing why the bridge server failed to start, with a Retry button.

— Named options and default state, limited scope detail. UI described but failure causes not detailed.

v0.8.0 v0.9.0

---

16 **Favorite-folder filtering in listFolders** NEW

56

Adds `favoritesOnly` filter and `isFavorite` field to `listFolders` to scope folder listings to starred folders.

— Named parameter and field, no usage example. v0.9.0

---

17 **Full-message read size and time limits** BREAKING

52

Full-message reads are now limited to 50 MiB per message with a 120-second deadline; batches share that deadline and a 50 MiB output budget.

— Concrete limits named but no workaround guidance. v0.10.0

┗──▷ BREAKING ON UPGRADE

! `createContact` , `updateContact` , and `deleteContact` are disabled by default after upgrading; re-enable them by turning on Allow contact writes in Options.

! `skipReview: true` is refused when there are no recipients, or when the message would exceed `mailnews.message_warning_size` (20 MiB by default).

! `skipReview` block now also governs scheduled drafts, and scheduling requires `sendMail` to be enabled.

! Message rows from `searchMessages` and `getRecentMessages` now cap at 3 recipients and 3 Cc addresses plus (+N more) by default; pass `fullRecipients: true` for complete lists.

! Profiles with restricted Tool Access must manually enable each of the nine new tools in Options — they start unchecked.

! Existing filters that move or copy messages to the Outbox (Unsent Messages) are silently skipped by `applyFilters` ; they must be disabled or deleted manually.

- ! Existing mail filters using 'is / isn't in address book' conditions are skipped by `applyFilters` when mail accounts are restricted and `Allow all address books` is off.
- ! `displayMessage` now refuses to display encrypted messages while `Allow MCP clients to read encrypted messages` is off (the default).
- ! Attachment paths on Windows that use short (8.3) names, reserved device names, or resolve to mapped network drives are now refused by the bridge — workflows supplying such paths will fail.
- ! Meetings (events with attendees) are read-only through MCP by default while `Block skipReview` is enabled — changing, deleting, or adding attendees to such events is blocked until the setting is turned off.
- ! Direct sends (`sendMail` with `skipReview`) now report success only after delivery fully completes; previously, rejected recipients or an unreachable server were reported as sent.
- ! Targeting a disabled calendar now returns an error instead of an empty list.
- ! Filter date conditions now accept only `YYYY-MM-DD` ; other date formats are rejected.
- ! A refused attachment (dot-directory path, `AppData` , macOS home `Library` , credential-like filename, UNC path, or Windows alternate data stream) now fails the entire send, draft, or reply — previously only the attachment was rejected.
- ! `getMessage` now returns a placeholder and `encryptedContentWithheld: true` for OpenPGP and S/MIME messages by default; any client that previously read encrypted message bodies must enable 'Allow MCP clients to read encrypted messages' in the Options page.
- ! MCP clients can no longer create, change, or run forward/reply filter actions by default; operators must enable `Allow automatic Forward/Reply filter actions` in the Options page.
- ! If `THUNDERBIRD_MCP_CONNECTION_FILE` points at a symlink or a file with permissions looser than owner-only, the bridge will refuse to start — point the variable at the real `connection.json` file with owner-only permissions.
- ! `applyFilters` now skips rules not marked for manual runs; filters previously triggered by `applyFilters` that lack the manual-run flag will no longer execute.

WAS THIS USEFUL? 👍 👎

MCP TypeScript SDK

Feedback ⓘ v2.4.0 NOTES ↗ RANK

The MCP TypeScript SDK provides TypeScript clients, servers, and transports for Model Context Protocol.

> The SDK's headline addition is Skills over MCP (SEP-2640), letting servers publish skill folders that clients can list, read, and verify, alongside opt-in OAuth scope forwarding on token refresh for Entra ID compatibility; the release also ships three breaking validation changes around `x-mcp-header` number parameters, `outputSchema` wrapping, and a stricter Ajv/fast-uri upgrade.

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

01 **Skills over MCP (SEP-2640)** NEW 98

Adds the Skills extension across four new subpath packages: `@modelcontextprotocol/server/ext/skills` exports `SkillsExtension` (attach(server) serves `skills/list`, `skills/get`, `resources/directory/read`, and a `skill://{+path}` resource template) plus SkillsExtension.load(files) returning a `SkillCatalog` with `entries` and reload(), the `SkillFiles` storage adapter with `memorySkillFiles`, `SkillSource`, `skillEntry`, and `SkillCatalogError`; `@modelcontextprotocol/server/ext/skills/node` exports loadSkills(folder) and nodeSkillFiles(folder) for Node.js file-system loading; `@modelcontextprotocol/client/ext/skills` exports `SkillsClient` with list() (`skipInvalid: true` keeps valid entries and returns the rest in `invalid`), get(), read(skill, path?) (verifies size, digest, and `SKILL.md` frontmatter, retrying with `cacheMode: 'refresh'` on a cache-hit mismatch and raising `SkillVerificationError`), and readDirectory(); `@modelcontextprotocol/core/ext/skills` exports schemas, types, constants, and `SKILL_LIMITS`, and on the 2026-07-28 protocol era the server sends `ttlMs` and `cacheScope` which `SkillsClient` requires and returns from list(), get(), and readDirectory(). A server publishes a skill folder with loadSkills(folder) and skills.attach(server); this works on both the 2025-11-25 and 2026-07-28 protocol revisions, though `attach` is marked experimental.

Publish a folder of skills from an MCP server so connected clients can list and invoke them.

```typescript
import { McpServer } from '@modelcontextprotocol/server';
import { loadSkills, skills } from
'@modelcontextprotocol/server';

const server = new McpServer({ name: 'my-server', version:
'1.0.0' });

await loadSkills('./skills');
skills.attach(server);
```

List available skills from a connected server, tolerating invalid entries, then read a specific skill's content with integrity verification.

```typescript
import { Client } from '@modelcontextprotocol/client';
import { SkillsClient } from
'@modelcontextprotocol/client/ext/skills';

const skillsClient = new SkillsClient(client);

// List skills, skipping invalid entries instead of throwing
const { skills, invalid } = await skillsClient.list({
skipInvalid: true });
console.log('valid:', skills, 'invalid:', invalid);

// Read a skill with size/digest/frontmatter verification
const content = await skillsClient.read(skills[0]);
```

List available skills from a server, tolerating invalid entries, then fetch and verify a specific skill's contents.

```typescript
import { SkillsClient } from
'@modelcontextprotocol/client/ext/skills';

const skillsClient = new SkillsClient(mcpClient);

// List skills, skipping invalid entries instead of throwing
const { entries, invalid } = await skillsClient.list({
skipInvalid: true });
console.log('valid:', entries, 'invalid:', invalid);

// Read and verify a skill (size, digest, SKILL.md frontmatter
checked)
const content = await skillsClient.read(entries[0]);
```

— Full API surface and runnable examples across all three releases `v2.4.0`

`@modelcontextprotocol/server@2.4.0` `@modelcontextprotocol/client@2.4.0`

02 **Breaking: x-mcp-header rejects number parameters** BREAKING

85

A tool parameter annotated with `x-mcp-header` whose JSON type is `number` is now rejected per the 2026-07-28 spec: clients exclude such tools from `tools/list` entirely on 2026-07-28 connections and send no `Mcp-Param-*` headers for them, and a direct call to such a tool now returns error code `-32020`. Change the parameter type to `integer` (z.int()) or remove the `x-mcp-header` annotation.

— Clear migration path but no code example given `v2.4.0`

`@modelcontextprotocol/server@2.4.0` `@modelcontextprotocol/client@2.4.0`

03 **Opt-in OAuth scope on token refresh** NEW

81

Adds opt-in `scope` on OAuth token refresh: implement scopeForRefresh(tokens) on your `OAuthClientProvider` to return the scope to send with the refresh request, satisfying authorization servers — such as Microsoft Entra ID, which rejects scope-less refreshes with `AADSTS90009` — that require `scope` on refresh.

Support Microsoft Entra ID (and other servers that require `scope` on refresh) by implementing `scopeForRefresh` on your OAuth provider.

```typescript
import { OAuthClientProvider } from
'@modelcontextprotocol/client';

class MyProvider extends OAuthClientProvider {
  scopeForRefresh(tokens) {
    // Return the original scope so Entra ID accepts the refresh
request
    return tokens.scope ?? 'openid profile';
  }
  // ... other required methods
}
```

Forward scope on OAuth token refresh to satisfy Microsoft Entra ID (AADSTS90009) when the client and resource share the same app registration.

```typescript
class MyOAuthProvider implements OAuthClientProvider {
  // ...existing required methods...

  scopeForRefresh(tokens: OAuthTokens): string {
    // Return the original scope so Entra ID accepts the refresh
    return tokens.scope ?? 'openid profile';
  }
}
```

— Named method and runnable provider example, but narrow single-method scope v2.4.0

@modelcontextprotocol/client@2.4.0

04 **Breaking: outputSchema no longer auto-wraps results**

BREAKING

75

A tool `outputSchema` whose root resolves to an object — including one derived from z.object(...).meta({ id }) or a `$ref` to an object entry in `$defs` / `definitions` — is now treated as an object schema: on 2025-11-25-era connections the schema and `structuredContent` are no longer wrapped in `{ result: ... }`, so a client must read `structuredContent` directly instead of `structuredContent.result`.

— Explains before/after behaviour but no runnable fix example v2.4.0

@modelcontextprotocol/server@2.4.0

05 **Breaking: stricter Ajv schema URI validation via fast-uri 3.1.8**

68

BREAKING

The bundled Ajv validator now uses `fast-uri` 3.1.8 (up from 3.1.0), which rejects a schema `$id` or `$ref` whose value has a colon-prefixed scheme that isn't a valid URI scheme (e.g. `my_tool:output`) or an invalid URN namespace (e.g. `urn:my_tool:output`); such schemas no longer compile, and any tool carrying one fails with an invalid-schema error on upgrade.

— Names the version bump and failure mode but no migration fix given `v2.4.0`

`@modelcontextprotocol/server@2.4.0` `@modelcontextprotocol/client@2.4.0`

THINNER COVERAGE BELOW

06 **ServerEvent params forwarding** IMPROVED 40

`ServerEvent` now accepts optional `params`, enabling a gateway to forward change notifications together with their `_meta`.

— Named field but no example or mechanism detail `v2.4.0`

└──▷ **BREAKING ON UPGRADE**

! `x-mcp-header` on a `number` parameter is now rejected per the 2026-07-28 spec: the client will not list such a tool and the server returns `-32020` on a call to it. Use z.int() or remove the annotation.

! On a 2025-11-25 connection, `outputSchema` derived from z.object(...).meta({ id }) no longer wraps results in `{ result }`: read `structuredContent` directly instead of `structuredContent.result`.

! The bundled Ajv validator now uses fast-uri 3.1.8, which rejects a `$id` or `$ref` with an invalid URI scheme (e.g. `my_tool:output`).

! The bundled Ajv validator now uses fast-uri 3.1.8; schemas whose `$id` or `$ref` value has a colon but an invalid URI scheme before it (e.g. `my_tool:output`) or an invalid URN namespace (e.g. `urn:my_tool:output`) no longer compile, and any tool that carries such a schema fails with an invalid-schema error on upgrade.

! A tool parameter annotated with `x-mcp-header` whose JSON type is `number` is now rejected by the server; on 2026-07-28 connections, clients exclude such tools from `tools/list` entirely and send no Mcp-Param-* headers for them. Change the parameter type to `integer` (z.int()) or remove the `x-mcp-header` annotation.

! A tool `outputSchema` whose root is a `$ref` to an object entry in `$defs` or `definitions` is now treated as an object schema: on 2025-era connections the schema and `structuredContent` are no longer wrapped in `{ result: ... }`, so a 2025-era client that previously read `structuredContent.result` must now read `structuredContent` directly.

! `x-mcp-header` on a `number` `-typed` tool parameter is now rejected; clients MUST exclude such tools from `tools/list`. Previously `number` was accepted — any tool using `number` with `x-mcp-header` will now fail validation.

! The bundled Ajv validator is built with fast-uri 3.1.8 (was 3.1.0); schemas whose `$id` or `$ref` values use a colon-prefixed prefix that is not a valid URI scheme (e.g. `my_tool:output`) or an invalid URN namespace (e.g. `urn:my_tool:output`) no longer compile and cause an invalid-schema error at runtime.

WAS THIS USEFUL? 👍 👎

mcp-server

Feedback ⓘ v2.2.7 NOTES ↗ RANK

BrowserStack's Official MCP Server

> BrowserStack's MCP Server significantly expanded its App Accessibility tooling across builds, reports and Automate sessions, added a feedback submission capability, and removed the older accessibilityExpert tool.

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

01 Expanded App Accessibility scanning and reporting IMPROVED

74

Accessibility scans now support WCAG version selection, best practices filtering, and a needs review categorisation. The server adds App Accessibility checks on native builds with App Automate links, can read accessibility issues from an Automate build or session link, and can list App Accessibility builds and reports (including reading manual reports) and set up App Automate runs for them. It adds routing of a11y checks, an app build summary with `compareWith`, exact paging, and reliable app build links. Separately, it fetches accessibility issues for builds, Workflow Analyzer reports, and dashboard links, each including a CSV link, and now waits for an accessibility report to finish processing before reading its issues to prevent incomplete result retrieval.

– Many named surfaces and behaviours, but no exact tool call syntax given v2.2.7 v2.2.5

THINNER COVERAGE BELOW

02 Removal of accessibilityExpert tool BREAKING

40

The `accessibilityExpert` tool has been removed from the MCP server.

– Names the removed tool but gives no migration path v2.2.7

03 Feedback submission capability NEW

20

Adds a feedback submission feature to the MCP server.

– Only a one-line description with no mechanism or interface detail v2.2.2

└──▷ **BREAKING ON UPGRADE**

! The `accessibilityExpert` tool has been removed.

WAS THIS USEFUL? 👍 👎

context7

Feedback ⓘ @upstash/context7-sdk@0.6.0 NOTES ↗ RANK

Context7 maintains up-to-date documentation and context for AI agents through an MCP server.

> Context7 expanded client support to ChatGPT on the web and gave the SDK's search results a unified cross-list ranking field.

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

01 **ChatGPT custom MCP server plugin** NEW 75

Context7 can now be added to ChatGPT on the web as a custom MCP server plugin, with OAuth authorization and `@` mention support to pull it into any chat.

Add Context7 as a custom MCP server in ChatGPT on the web to pull live library docs into any chat via `@` mention.

📍 1. In ChatGPT, open ChatGPT Plugins and select the plus button › Add custom MCP server. 2. Fill in the server details, select `I understand and want to continue`, then `Create as a plugin`. 3. Find the plugin in your plugins list, install it, and complete the one-time OAuth authorization. 4. Start a new chat, type @ in the prompt box, and select Context7.

– Includes exact step-by-step setup instructions product docs

02 **Rank field in SDK search results** IMPROVED 65

The `@upstash/context7-sdk` search() method now returns a `rank` field (1-based integer) on each code and info snippet, reflecting its position in a shared reranked order across both `codeSnippets` and `infoSnippets`.

– Names exact field and behavior but no usage example @upstash/context7-sdk@0.6.0

WAS THIS USEFUL? 👍 👎

claudexor

Feedback ⓘ v4.7.0 NOTES ↗ RANK

Multi-harness control plane for Claude Code, Codex, Cursor, and OpenCode: quota-aware rotation across multiple Claude/Codex subscriptions, shared thread context, and cross-model review.

> Claudexor's biggest change this window is v4.0's consolidation of all local state into a single SQLite database (Node 24.15+ required, no downgrade to 3.x), alongside new capability-aware model routing with routing-evidence APIs, image creation/editing via a connected Codex account, scoped OS-process cleanup with run-scope visibility, and expanded quota/spend observability for Codex subscriptions and Cursor.

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

01 Capability-aware model routing and routing evidence

BREAKING

96

Adds `captureRoutingEvidence=true` on `POST /v2/model-operations` to capture per-request routing receipts bound to the idempotency key, and `includeCapabilities=true` on `GET /v2/accounts/routing-evidence` to expose per-account model capability marks with episode start times. Pre-generation refusal responses now carry `modelMarks` and `alternativeRemains` fields, and `model_capability_refused` refusals add `retryOnOtherAccount` and `retryOfOperationId` fields offering one correlated retry on another account. `model_capability_refused` is a new distinct error code for recognized Codex model-not-found/not-supported refusals, replacing the former `invalid_request` or `model_unavailable` codes — clients branching on those codes must update. Capability-aware auto routing now prefers accounts whose live catalog lists the requested model, and an account that returns a recognized `model_capability_refused` gets a temporary four-hour capability mark suppressing traffic for that model; `preferredProfileId` no longer wins over a listed account or over its own capability mark or recent refusal (see `docs/MODEL_CAPABILITY_ROUTING.md`).

Capture routing evidence for a model operation to audit which account served the request and whether any capability marks were applied.

```
$ curl -X POST 'https://localhost:<port>/v2/model-operations?
captureRoutingEvidence=true' \
  -H 'Idempotency-Key: <idempotency-key>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "<model>", "messages": [{"role": "user",
"content": "Hello"}]}'
```

Inspect live capability marks across all accounts to see which models have been temporarily suppressed and whether alternatives remain.

```
$ curl 'https://localhost:<port>/v2/accounts/routing-evidence?
includeCapabilities=true'
```

— Endpoints, fields, error code and behavior change all named with mechanism. v4.5.0

02 **Scoped OS-process cleanup and run-scope visibility** NEW

96

Adds `summary.processScope` on `GET /v2/runs/:id`, carrying the compact scope record once a run is terminal — explicit `null` when unknown, never reported as clean, and always `null` on the run list (read the detail endpoint). Adds `daemon/run-scopes.json` (`{ "schema_version": 1, "entries": [...] }`) as a documented contract file readable by a host's emergency stop while the daemon is dead, and `final/process_scope.yaml` written at run termination to record survivors, unreadable members, and reused process groups. Scoped OS-process cleanup now triggers at every terminal state (success, failure, cancel, timeout) and at the next start after a crash, bounded by a single 5 s absolute settlement budget covering seed refinement, census, identity and table reads, and signals; incomplete work is disclosed as `incomplete`, never signalled late. Coverage is disclosed per platform as `complete` on Linux with a readable census, `not_provable` on macOS (SIP hides Apple platform-binary environments), and `tree_only` on Windows.

Poll a finished run to inspect whether any OS processes survived the settlement — `null` means scope is unknown, not clean.

```
$ curl -s https://localhost:<port>/v2/runs/<run-id> | jq
'.summary.processScope'
```

Read the daemon's emergency-stop contract file to enumerate active run scopes when the daemon is not running.

```
$ cat ~/.claudexor/daemon/run-scopes.json | jq '.entries'
```

Inspect which processes survived settlement or were unreadable after a run completes.

```
$ cat <run-artifact-dir>/final/process_scope.yaml
```

— Names endpoint, files, schema and platform-specific values with runnable examples.

v4.1.1

03 **Per-turn auxiliary writable roots** NEW 91

Native workspace-write turns can now be granted access to caller-specified external directories (e.g. download caches, verification scratch space) on a per-turn basis via the `--auxiliary-writable-root` CLI flag and `execution.auxiliaryWritableRoots` config key. The `accessControl.auxiliaryWritableRoots` negotiated capability lets harnesses detect support at the protocol level; HTTP callers can clear inherited roots with `auxiliaryWritableRoots: []` in a complete execution replacement group, while omitting the field in `continueFrom` inherits the prior group's roots. Admission rejects paths overlapping source/execution trees, protected engine/profile state, or macOS directory aliases, and the grant adds no socket or process-enumeration rights.

Set auxiliary writable roots in config so every workspace-write run in this project can reach the same cache directories.

```yaml
execution:
  auxiliaryWritableRoots:
    - /tmp/dl-cache
    - /tmp/verify-scratch
```

— Full flag, config key, capability name and rejection rules given. v4.7.0

04 **Quota freshness and unobserved Codex spend reporting** BREAKING 85

Adds `GET /v2/quota?view=resources_freshness` combining per-constraint freshness, typed account resources, and a required `availability_scope` echo with `basis: "quota_block"`, the exact supplied model, and the effective request kind. Adds `controllableSpendUsd` and `budget_summary.controllable_cash_usd` fields to represent the cash a finite cap governs when Codex subscription spend is unobserved; in that case `spendUsd` and `budget_summary.spend_usd` / `cash_usd` are null and `debit` declares `"unobserved"` — verified Codex subscription work now

reports spend as unknown instead of `$0`, so callers reading those fields as zero will see null. `claudexor inspect` now prints `budget: spend=unknown controllable=<usd>` for these runs.

— Named endpoint, fields and CLI output with explicit before/after. v4.3.0

05 **Managed Codex install targeting flags** NEW 83

Adds `--target local` and `--release-tested` flags to managed Codex installs: `--target local` keeps a runnable managed copy at or above the release-tested baseline, and `--release-tested` pins exactly the baseline version. A registry failure or a `latest` older than the baseline (`latest_older_than_tested`) refuses instead of falling back, and install receipts gain a `pinnedVersion` field naming the selected install target.

Install the release-tested baseline Codex version into the managed local toolchain — useful when you want a fully reproducible harness version pinned to what Claudexor tested against.

```
$ claudexor harness install codex --target local --release-tested --yes --json
```

— Exact flags, error code and field with runnable install command. v4.4.0

06 **Run control via HTTP/MCP** NEW 80

Adds `idempotencyKey` to MCP controller run launches for idempotent one-shot runs, four new MCP tools — `claudexor_run_wait` for bounded event replay with lifecycle and pending-interaction snapshots (stopping observation never cancels the run), `claudexor_run_message`, `claudexor_run_continue` which resolves omitted controls, access included, to new-run defaults, and `claudexor_run_event_read` which reads oversized events in bounded parts while reporting exact redacted bytes and hash — and HTTP/MCP controller support to read, set, extend, or disable a live run's soft/hard timer with revision checks, where `set` preserves the current mode unless a mode is supplied and omitting `maxSeconds` adds no deadline.

— Names all four tools and the idempotency key with behavior detail. Describes the operations and defaults but no exact endpoint/flag named.

v4.4.0 v4.3.0

07 **Durable store ID in handshake** NEW 75

`POST /v2/handshake` adds a `storeId` field returning the durable engine store ID when the client sends `X-Claudexor-Store-Identity: 1`; clients that omit the header get the unchanged prior response shape.

Retrieve the durable engine store ID from a running daemon to correlate clients across sessions.

```
$ curl -s -X POST http://localhost:<port>/v2/handshake -H 'X-Claudexor-Store-Identity: 1' | jq .storeId
```

— Exact endpoint, field and header with runnable example. v4.4.0

08 **Unified SQLite storage engine** BREAKING 72

Consolidates commands, conversations, resource metadata, and retained events into a single SQLite database, with startup importing legacy journals once while preserving original bytes, run artifacts, and accounts. `status` exposes storage, event-loop, and admission facts. Logical recovery isolates a project; physical corruption triggers whole-store recovery with earlier generations retained as evidence. Live event streams expose retained rows, so a superseded row may not be delivered. Node.js 24.15 or newer is now required, Claudexor.app 4.0 is required for desktop engine updates (an older app keeps its working engine but prompts for an update), and the migrated storage format is forward-only — downgrading to 3.x is unsupported.

— Describes the migration and requirements but no command to inspect it. v4.0.0

09 **Daemon status latency and commit telemetry** IMPROVED 70

Daemon status now reports per-route request latency (p50, p99, and max for lifetime and last-loop window, keyed by HTTP method and operation-catalog template) and the 32 longest labelled synchronous sections since start (store `BEGIN` / `COMMIT`, external writes, run detail and summary projections, scope persistence) with duration, time, and lifetime `COMMIT` p50/p99. It also exposes native-COMMIT counts and durations since the engine store opened, without any durability or admission-policy change.

— Names exact metrics and status surface but reached via status command only. v4.4.0 v4.1.1

10 **Run retry and legacy upload failure handling** IMPROVED 65

An Exact Retry of a run whose files were retired is now refused, with Run Again offered as the next action. Legacy upload bindings that cannot be read or parsed after a SQL miss now answer HTTP 503 `legacy_upload_binding_unavailable` instead of minting a new upload; while the legacy directory is unreadable, every SQL miss is refused, including new keys.

— Names exact error code and behavior but both are edge-case refusals. v4.4.0

11 **Account resource and spend reporting** NEW 65

Exposes independently dated balances, spending, and reset inventory alongside included quota via the control API, CLI, MCP, and native interface, including explicit Codex and Claude resets distinct from Claude-granted resets and five-hour session refills, with original request identity preserved across uncertain replies so provider outcome stays separate from readback and unknown values remain unknown rather than defaulting; adds independently dated personal and team balances and spending for Cursor, with optional `scope`, `usage_kind`, `period`, and `fee_included` fields, sourced from Cursor's internal `usage-summary` dashboard via the existing refresh demand and pacing.

— Names fields and data source but no direct command shown. Names the surfaces exposed but no exact endpoint or flag given.

v4.3.0 v3.25.1

---

12 **Scoped-home environment variables for Playwright cache** NEW

65

Adds `CLAUDEXOR_HOME_SCOPED` and `CLAUDEXOR_OPERATOR_HOME` environment variables to notify scoped-home sessions of their scope and point them at the operator's Playwright browser cache, and adds `PLAYWRIGHT_BROWSERS_PATH` support so an explicit browser cache path (including `0`) wins over the default for scoped-home sessions.

— Names all three environment variables with their effect. v4.1.1

---

13 **Speed preference frozen per routed lane** BREAKING

60

Speed preference is now frozen per routed lane in `context/processing_intent.yaml` across retries, Exact Retry, account rotation, and in-run continuation, instead of adopting another account's setting; without an explicit preference, these now keep the speed the first Codex account chose.

— Names the config file and scope but no direct override command. v4.3.0

THINNER COVERAGE BELOW

14 **Codex workspace-write default to native auto-review** BREAKING

58

Codex workspace-write runs now use native `on-request` with `auto_review` by default, meaning the vendor's automatic reviewer can approve individual actions outside the base workspace-write sandbox without a controller; unanswered denial handbacks wait up to 15 minutes before a benign decline.

— Explains the behavior change and timeout but no config toggle named. v4.3.0

---

15 **Image creation and editing via Codex account** NEW

58

Adds image creation and editing through a connected ChatGPT/Codex account, surfacing exact edit bytes, usable partial images, requested versus observed model, and unknown cost as explicit typed facts. Image-only quota windows are now exposed separately so they stay visible without blocking text work, and unrecognised Codex full-read/rollout buckets are no longer treated as quota constraints, with notification payloads retaining diagnostics.

— Explains the feature and facts surfaced but no endpoint/command shown. v4.1.0

16 **Truncated output disclosure across harnesses** NEW 54

Adds `output_drain: "truncated"` disclosure on completed events in Codex, the shared CLI loop (Claude, Cursor, Antigravity, OpenCode), and ACP when a bounded post-exit drain closes before all output is consumed; gates store the equivalent as `output_truncated`.

— Names the exact field and value across harnesses, no direct command. v4.1.1

17 **Daemon load benchmarking script** NEW 48

Adds `scripts/daemon-load-bench.mjs` for comparing an exact base against the working tree in alternating trials inside temporary roots.

— Names the script path but no usage example given. v4.4.0

18 **Antigravity headless permission-denial detection** IMPROVED 47

Antigravity now recognizes native headless permission auto-denial on stderr as typed `permission_denial` evidence (`capability_refused` / `permission_denied` , resumable cause `permission_denied`) and ends the attempt without retrying or escalating.

— Names the typed evidence codes but is an internal detection change only. v4.3.0

19 **Shared queue for model calls and long sessions** NEW 43

Model calls and long sessions now share one queue with optional separate caps; omitting a cap means unlimited, and an existing explicit global cap remains global.

— Explains the queue behavior but names no config key to set caps. v4.0.0

20 **Codex Fast counted as included subscription service** IMPROVED 41

Ordinary Codex Fast (`fast` / `priority`) on a verified ChatGPT subscription now counts as included subscription service and stays eligible under `paid_fallback: never` and finite(0).

— Names the exact setting values but is a narrow eligibility fix. v4.3.0

21 **Codex model catalog inputModalities fallback** BREAKING 41

Codex model catalog `inputModalities` declarations that are missing, empty, or partly invalid now read as unknown (`[]`) rather than returning a filtered subset.

— Names the exact field and new fallback value only. v4.3.0

22 **Readonly Agent startup on broad source scopes** IMPROVED 40

Readonly live-directory Agents now skip write-oriented baseline inventory and output capture, unblocking startup on broad source scopes containing sockets and preventing source files from being copied into attempt artifacts.

— Explains behavior change but no config surface to act on. v4.5.0

23 **Ad-hoc signing of local macOS app bundles** IMPROVED 30

Local macOS app bundles are now ad-hoc signed and strictly verified before a ZIP or DMG artifact is produced.

— Names the mechanism but no flags or commands given. v4.6.2

24 **Cancellation-before-dispatch reporting** IMPROVED 25

Cancellation before provider dispatch now reports that no request was sent.

— One-line behavior note with no mechanism detail. v4.0.0

┗──▷ BREAKING ON UPGRADE

! Recognized Codex model-not-found and not-supported refusals now return `model_capability_refused` instead of `invalid_request` or `model_unavailable` ; clients that branch on those former codes must be updated.

! `preferredProfileId` no longer wins over a listed account or over its own capability mark or recent refusal; history bound to that account keeps it first over a catalog miss only while it has neither — see `docs/MODEL_CAPABILITY_ROUTING.md` .

! Verified Codex subscription work now reports spend as unknown instead of `$0` ; `spendUsd` and `budget_summary.spend_usd` / `cash_usd` are null, and `debit` is `"unobserved"` — callers reading those fields as zero will see null.

! Codex workspace-write runs now use native `on-request` with `auto_review` by default; the vendor's automatic reviewer can approve individual actions outside the base workspace-write sandbox without a controller.

! Without a speed preference, retries, Exact Retry, account rotation, and in-run continuation now keep the speed the first Codex account chose, frozen in `context/processing_intent.yaml` , instead of adopting another account's setting.

! Codex model catalog `inputModalities` is now treated as unknown (`[]`) when the declaration is missing, empty, or partly invalid, rather than returning a filtered subset.

! Node.js 24.15 or newer is required; older Node versions are no longer supported.

! Claudexor.app 4.0 is required for desktop engine updates; an older app retains its working engine but will prompt for an app update.

! The migrated storage format is forward-only: downgrading to 3.x is unsupported.

WAS THIS USEFUL? 👍 👎

davinci-resolve-mcp

Feedback ⓘ v4.15.0 NOTES ↗ RANK

MCP server integration for DaVinci Resolve Studio

> This window's biggest changes are new timeline actions for source-range slipping and keyframe-override detection, a full subtitle-preset discovery/application pipeline in timeline_ai, and expanded scripting-availability diagnostics via doctor.py, alongside a breaking change to the default `timeline_frame capture` method; smaller additions touched render settings, rename planning, high-priority mode, and Word Highlight styling.

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

01 Subtitle preset discovery and application in timeline_ai NEW

98

Adds timeline_ai(action="list_title_presets") to inventory installed factory Fusion title and subtitle presets, returning local archive, category, and full template ID, and accepting `templates_archive` to point at a custom installation. Adds timeline_ai(action="set_subtitle_preset") to apply an installed subtitle preset with literal style inputs (font, size, position) to a subtitle track, writing a new timeline revision while keeping the original and restoring on failure; dispatcher-level `dry_run` / `dryRun` coercion previews the operation without importing. Results include `verified_components` (track counts, item names and bounds, media IDs, preset controls) and `unverified_components` (markers, grades, other Fusion comps, audio), and staging exports are removed on success or retained for diagnosis on failure as `staging_retained` / `staging_directory`. Adds `scripts/edit_subtitle_timeline.mjs` (backed by `resolve-advanced/server/subtitle-timeline.mjs`) to inspect an exported subtitle track or write a new file referencing a bundled preset (Word Highlight , Lollipop) or a native subtitle-template.

Discover all installed subtitle presets and their template IDs before deciding which to apply, pointing at a custom archive.

```
$ timeline_ai(action="list_title_presets",
templates_archive="/path/to/custom/archive")
```

— Full pipeline with runnable example and named result fields v4.12.0

02 Reversible high-priority mode via set_high_priority NEW

93

Adds a new set_high_priority(high) granular tool (alias `high_priority`, default `true`) that calls SetHighPriority(bool) on DaVinci Resolve 21.1+, making priority changes reversible — `high=false` restores normal priority; pre-21.1 builds keep the no-argument call and refuse `high=false`.

Restore normal script execution priority after a high-priority render pass on Resolve 21.1+ without restarting the server.

```
$ resolve_control set_high_priority high=false
```

— Runnable command with version-gated behaviour detailed v4.13.0

---

03 **Scripting-availability diagnostics via doctor.py** IMPROVED

91

Adds `scripts/doctor.py` Python-environment checks (in `scripts/doctor_python_env.py`) covering Python bitness, version compatibility with Resolve, and interpreter identity across Claude Desktop and Codex configs, each reporting WARN or FAIL with a `Fix:` line and a `fix` key under `--json`. The `SCRIPTING_UNAVAILABLE` error state gained a `scripting_preference_local` field (true, false, or null) read from Resolve's `config.dat` `System.Scripting.Mode`, surfacing a Quit and relaunch DaVinci Resolve hint when the preference is already Local, and its remediation now points at `scripts/doctor.py` when Studio is already set to Local.

Diagnose whether the server is running on the wrong Python interpreter before a scripting session fails with `SCRIPTING_UNAVAILABLE`.

```
$ python scripts/doctor.py --json
```

— Runnable diagnostic command and exact config fields named v4.15.0

---

04 **AudioFormat and UseUniqueFilenames render settings** NEW

81

Adds `AudioFormat` and `UseUniqueFilenames` to the compound render-settings whitelist, making them accepted across `set_settings`, `safe_set_render_settings`, `prepare_render_job`, and `set_render_settings`; unrecognised values are rejected with `INVALID_RENDER_SETTING`.

Include audio format and unique filenames in a compound render job to avoid overwriting earlier exports in the same output folder.

```
$ prepare_render_job AudioFormat="wav" UseUniqueFilenames=true
```

— Runnable example with named APIs and error code v4.13.0

05 **Changed default capture method for timeline_frame** BREAKING

80

`timeline_frame capture` now defaults `quality="frame"` to use `Project.ExportCurrentFrameAsStill` (no render settings, no page switch), with `quality="render"` retained for explicit single-frame rendering; `preview` and `full` now also use the export route. Workflows that relied on the render path at the default quality must now pass `quality="render"` explicitly.

— Exact migration flag given with clear before/after v4.15.0

06 **Case-sensitive comparison mode for rename_plan** NEW

78

Adds `caseSensitive` parameter (default `false`) to `rename_plan`, enabling exact byte-for-byte name comparison for case-sensitive volumes; the result object reports the mode used as `caseSensitive`.

When planning renames on a case-sensitive Linux volume, pass `caseSensitive: true` to treat `Clip.mov` and `clip.mov` as distinct files and avoid false collision suppression.

```
$ media(action="rename_plan", caseSensitive=true)
```

— Runnable example and default value given v4.11.0

07 **Timeline clip source-range replacement** NEW

75

Adds `timeline replace_source_range` action to slip a clip to a new source range via a temporary-track rebuild, with `dry_run`, confirm-token gating, archiving, and `unlock_track=true` for locked tracks.

— Mechanism and flags named, no runnable example given v4.15.0

08 **Playback-active guard on page-switch actions** NEW

73

Page-switch actions (`resolve_control open_page`, `restore_state`, `gallery_stills grab_and_export`, and capture routes) now detect a moving playhead and refuse with `PLAYBACK_ACTIVE`; `allow_during_playback=true` overrides the refusal.

— Named actions and override flag, no example v4.15.0

09 **Keyframe override detection on timeline items** NEW

65

Adds `timeline_item probe_keyframe_override` action that detects whether a property write actually reaches the render by exporting frames before and after a nudge,

returning `override`, `writes_render`, or `inconclusive`.

— Clear mechanism and return values, no example v4.15.0

---

10 **Shadow and alignment controls in Word Highlight macro** NEW

65

Adds `shadowEnabled`, shadow color, `shadowOffset`, softness, `horizontalAlign`, and `verticalAlign` inputs to the Word Highlight subtitle macro; the centred reference value for alignment is `3`, and a dry run lists all `editableInputs`.

— Named inputs and reference value, no example v4.14.0

---

11 **Expanded text-node coverage for title/subtitle text** IMPROVED

65

Expands `timeline.set_title_text` / `get_title_text` Fusion-comp fallback to find literal text on TextPlus, sText, Text3D, and MultiText nodes via `TextN.StyledText` in native `TextOrder`.

— Named node types and property path, no example v4.12.0

---

12 **Pixel-based pan/tilt for set_transform** NEW

60

Adds `pan_px` and `tilt_px` parameters to `set_transform`, computing pixel offsets from fitted image size and reporting the scale factors used.

— Named params and behavior, no example v4.15.0

THINNER COVERAGE BELOW

---

13 **Path and name redaction for QA reports** NEW

40

Adds `src/utils/qa_privacy.py` to replace host paths and project/timeline names in synthetic title-library QA reports.

— Thin description with no usage detail v4.12.0

└──▷ BREAKING ON UPGRADE

! The default for `timeline_frame capture` changed: `quality="frame"` (and `preview` / `full`) now uses `Project.ExportCurrentFrameAsStill` instead of the single-frame render path. Workflows that relied on the render path at the default quality must now pass `quality="render"` explicitly.

WAS THIS USEFUL? 👍 👎

pi-mcp-adapter

Feedback ⓘ v6.0.0 NOTES ↗ RANK

Token-efficient MCP adapter for Pi coding agent

> pi-mcp-adapter's biggest window update adds a full Executor integration — app tools exposed as named Pi tools, adapter-level approval handling, automated account sign-in, and longer timeouts — alongside `scriptTools` for saving workflows as callable tools, while removing the legacy jev/mcpScript scripting system in favor of Pi's codemode. Smaller releases added trusted-project server startup, per-server structured-content and roots controls, MCP log persistence, a gh-CLI-based GitHub auth preset, and a configurable MCP panel keybinding.

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

01 scriptTools for saving workflows as callable tools NEW

95

The adapter config gains `scriptTools` to save a script file as a named Pi deferred tool (`mcp__script__<name>`) with its own description and input schema, callable via `tool_search` or mcp({ search }); `McpConfig` gains `scriptTools` typed by the new `ScriptToolDefinition` in `pi-mcp-adapter/types` . A new `settings.allowScriptToolSave` controls whether the model can call mcp({ action: "save-script", args: { name, description, code, inputSchema } }) to propose a script tool, with Pi showing diffs and opening the script in its editor for review before writing to `mcp-scripts/<name>.js` .

Define a reusable summarise-PR workflow as a named tool so the model can find and call it without rewriting the script each time.

```json
// ~/.config/mcp/mcp-adapter.json (adapter config)
{
  "scriptTools": [
    {
      "name": "summarise-pr",
      "description": "Fetch a GitHub PR and return a one-
paragraph summary.",
      "file": "mcp-scripts/summarise-pr.js",
      "inputSchema": {
        "type": "object",
        "properties": {
          "pr_url": { "type": "string" }
        },
        "required": ["pr_url"]
      }
    }
  ]
}
```

Let the model propose saving a working codemode script as a permanent tool, with Pi asking you to review diffs and edit before anything is written.

```javascript
// In a Pi codemode script, after the agent has a working
solution:
mcp({ action: "save-script", args: {
  name: "summarise-pr",
  description: "Fetch a GitHub PR and return a one-paragraph
summary.",
  code: "const { pr_url } = input;\n// ... fetch and summarise
logic ...",
  inputSchema: {
    type: "object",
    properties: { pr_url: { type: "string" } },
    required: ["pr_url"]
  }
} });
```

Disable the model's ability to propose new script tools in a locked-down project config.

```json
// ~/.config/mcp/mcp-adapter.json
{
  "allowScriptToolSave": false
}
```

— Full config keys, tool naming, and review workflow all specified. v6.0.0

02 **Executor and MCP tools exposed as named Pi deferred tools** NEW

93

Each Executor app tool is now exposed as a Pi deferred tool named `mcp__<server>__<app>_<tool>` (up to 1,000 per Executor server), and every cached MCP tool is now exposed as a Pi deferred tool named `mcp__<server>__<tool>` on Pi 0.99+; both are matchable by `approveTools` rules such as `github.*` and discoverable via `tool_search`, searchTools(), and mcp({ search }), with `--exclude-tools 'mcp__*'` removing the MCP ones for a session. `ServerCacheEntry` gains `executorApps`, `ToolMetadata` and `DirectToolSpec` gain `executorPath`, and `pi-mcp-adapter/types` exports `CachedExecutorAppTool`.

— Names naming scheme, limit, type additions, and discovery APIs. Names tool naming scheme and the `--exclude-tools` flag to disable it.

v6.0.0

03 **Session-root substitution for MCP servers** BREAKING

91

MCP servers that request roots now automatically receive the Pi session directory, enabling file and code-search servers (such as the official filesystem MCP server) to scope their work to the current project. The per-server config key `"roots": false` opts a server out of this automatic session-directory substitution, keeping its originally configured args; this is a breaking change for the official filesystem MCP server, whose directories in `args` are now replaced with the Pi session directory by default.

Keep a filesystem MCP server's originally configured directories instead of having them replaced by the Pi session root.

```json
{
  "mcpServers": {
    "filesystem": {
      "command": "npx",
      "args": ["-y", "@modelcontextprotocol/server-filesystem",
"/home/user/projects"],
      "roots": false
    }
  }
}
```

— Names config key, affected server, and breaking before/after. v5.4.0

04 **Legacy script execution (jev/mcpScript) removed** BREAKING

90

The `mcpScript` tool is removed; add `"+codemode"` to `defaultTools` in Pi's settings to script MCP calls with Pi's `codemode` instead. `settings.scriptMode` and `settings.scriptSkill` no longer do anything (configs still load with a notice naming the file and keys). The `settings.jev` keys `scriptEvaluation`, `maxEvaluationsPerScript`, `maxEvaluationBytesPerScript`, and `maxEvaluationTokensPerScript` no longer do anything, and jev.evaluate(), the bundled `mcp-scripting` skill, the `examples/jev-*.mjs` recipes, and the `quickjs-wasi` dependency are all removed. `McpSettings` loses `scriptMode` and `scriptSkill`; `McpSettings.jev` loses the four evaluation keys; `JevErrorCode` loses `"budget_exhausted"`; `JevBudget` is gone from `ResolvedJevSettings`. The new Executor and `scriptTools` features require Pi 0.99 or later.

— Enumerates every removed key, tool, dependency, and the migration path. v6.0.0

05 **GitHub setup preset authenticates via gh CLI** BREAKING

86

The `/mcp-adapter setup` GitHub preset now authenticates with `gh auth token` instead of MCP OAuth, making GitHub connections work out of the box. The old OAuth-based entry no longer works; remove the `github` entry from your config and re-add GitHub via `/mcp-adapter setup`, which requires `gh` installed and signed in via `gh auth login`.

— Exact command, migration steps, and prerequisite named. v5.3.2

06 **Per-server structured-content suppression** NEW

81

The per-server config key `"appendStructuredContent": false` shows the model only a tool result's `content`, omitting the `structuredContent` payload and saving context

on every call.

Stop a server that returns a verbose machine payload in `structuredContent` from burning context on every tool call.

```json
{
  "mcpServers": {
    "my-api-server": {
      "command": "npx",
      "args": ["my-api-mcp"],
      "appendStructuredContent": false
    }
  }
}
```

— Names exact config key, file, and effect. v5.4.0

07 **Unattended startup for trusted project MCP servers** NEW 80

The `pi --mcp-project-servers allow` flag starts a trusted project's MCP servers without an approval prompt, for that process only — untrusted projects stay blocked.

Run Pi in a new git worktree from a CI or automation script without stalling on an MCP approval prompt.

```
$ pi --mcp-project-servers allow
```

— Exact flag and scope given, directly runnable. v5.4.0

08 **Configurable keybinding to open MCP panel** NEW 78

The `mcp.panel.open` command can be bound in Pi's `keybindings.json` (e.g. `{ "mcp.panel.open": "alt+m" }`) to open the MCP panel with any key without clearing the prompt you are typing; no default key is assigned.

Bind a key to open the MCP panel so you can toggle servers on or off mid-session without losing the prompt you are composing.

```json
{ "mcp.panel.open": "alt+m" }
```

— Exact command name, file, and example binding given. v5.3.0

09 **MCP server log persistence to mcp.log** NEW 73

Server log messages are now appended to `~/.pi/agent/mcp.log` (rotating to `mcp.log.1` past 5 MB) instead of being dropped, matching the file and format Pi's built-in MCP uses.

— Exact path, rotation size, and format given. v5.4.0

10 **Executor call handling improvements** IMPROVED 60

Executor `execute` and `call` now get a 6-minute timeout unless the server sets `requestTimeoutMs`, replacing the previous 60-second default, and the adapter now intercepts Executor approvals and questions so Pi's own dialogs ask the user directly, refusing any `resume` call that comes from the model.

— Gives exact before/after values but no further mechanism. Explains mechanism and model-side restriction, no config surface.

v6.0.0

THINNER COVERAGE BELOW

11 **Automated Executor account sign-in** NEW 55

When a single `call` fails because an account is not connected, Pi now fetches the sign-in link itself, asks before opening it in the browser, and retries the call once — without the link ever reaching the model.

— Full behavioural flow described, no named flags. v6.0.0

12 **Startup logging for conflicting server URL configs** IMPROVED 55

When config files conflict on a server's `url` — causing credentials to be silently dropped — the adapter now logs at startup which server, which settings, and which two files are involved.

— Describes diagnostic output but introduces no new config surface. v5.3.2

13 **MCP panel shows tools for non-cached servers** IMPROVED 50

The MCP panel now shows tools for connected servers whose tool lists cannot be cached (such as GitHub), pulling them from the in-session connection instead of showing `(not cached)`.

— Explains behaviour change but no config surface. v5.3.2

14 **MCP sampling uses Pi's thinking level** IMPROVED 50

MCP sampling requests now use the current Pi thinking level, so servers that require a reasoning-capable model no longer fail with 'Reasoning is mandatory for this endpoint and cannot be disabled'.

— Names the exact error fixed, no user-facing action needed. v5.3.2

15 **User-Agent header on Parallel Search preset** IMPROVED

40

The Parallel Search preset in `/mcp setup` now sends `User-Agent: pi-mcp-adapter` to identify traffic to Parallel, with no user or install details.

— Names exact header value but limited broader context. v5.3.0

16 **MCP support in Pendant VS Code extension** NEW

30

MCP now works in the Pendant VS Code extension and other hosts that freeze `ctx.ui.theme`.

— Names the fix but gives no further scope. v6.0.0

┗──▷ BREAKING ON UPGRADE

! The `mcpScript` tool is removed. Add `"+codemode"` to `defaultTools` in Pi's settings to script MCP calls with Pi's `codemode` instead.

! `settings.scriptMode` and `settings.scriptSkill` no longer do anything; configs that set them still load and display a notice naming the file and the keys.

! The `settings.jev` keys `scriptEvaluation`, `maxEvaluationsPerScript`, `maxEvaluationBytesPerScript`, and `maxEvaluationTokensPerScript` no longer do anything.

! jev.evaluate(), the bundled `mcp-scripting` skill, the `examples/jev-*.mjs` recipes, and the `quickjs-wasi` dependency are removed.

! `McpSettings` loses the `scriptMode` and `scriptSkill` fields; `McpSettings.jev` loses `scriptEvaluation`, `maxEvaluationsPerScript`, `maxEvaluationBytesPerScript`, and `maxEvaluationTokensPerScript`; `JevErrorCode` loses `"budget_exhausted"`; `JevBudget` is gone; `ResolvedJevSettings` loses the same four fields.

! The new Executor and `scriptTools` features require Pi 0.99 or later.

! The official filesystem MCP server now has the directories in its `args` replaced with the Pi session directory by default; add `"roots": false` to its server entry in your config to keep the original directories.

! The old `/mcp-adapter setup` GitHub preset used MCP OAuth; that entry no longer works. Remove the `github` entry from your config and re-add GitHub via `/mcp-adapter setup`. Requires `gh` installed and signed in via `gh auth login`.

! A project-server approval saved under a differently-capitalized Windows path (e.g. `C:\Users\Me` vs `c:\users\me`) will prompt 'Allow project MCP server?' once more after this update before the normalized approval takes effect.

WAS THIS USEFUL? 👍 👎

gram

Feedback ⓘ tunnel@0.2.0 NOTES ↗ RANK

Gram provides MCP servers and AI-agent tools from APIs with access controls and observability.

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

01 **Workload identity authentication for device agents**

UPDATED INTEGRATION UPDATED SUPPORTED PLATFORM NEW 95

Headless device agents can authenticate to the device-agent API as a workload by exchanging a signed identity assertion at `/oauth/workload/device-agent/{organization_id}/token` for a 15-minute bearer token; `Authorization: Bearer` is accepted by `agent.getPlugins`, `hooks.ingest`, and OpenTelemetry log/metric endpoints, with every `hooks.ingest` request required to name its project via Gram-Project and the agent holding `org:hooks_ingest` for that project. New `workloadIdentities.createDeviceJoinToken` and `workloadIdentities.registerDevice` endpoints enroll headless boxes (e.g. a one-off VPS with no platform identity) as workloads, requiring `org:device_agent_sync` on the agent policy and `agent:authorize` on the owning admin. Device agents polling with an agent API key now also receive `mcp_servers` in the response — Speakeasy-hosted MCP servers for assigned plugins — written into each tool's configuration under a stable `speakeasy-<slug>` name; each entry includes token endpoint, assertion audience, and resource fields for the workload identity exchange, the list is always present (empty for people), and changes update the poll's ETag.

Enroll a headless VPS as a workload device so it can authenticate to the device-agent API without a static API key.

```
$ # Step 1: Admin creates a single-use join token for the agent
workloadIdentities.createDeviceJoinToken({ agent_id:
'<agent_id>' })

# Step 2: The device registers its public key using the join
token
workloadIdentities.registerDevice({ join_token: '<token>',
public_key: '<device_public_key>' })

# Step 3: Device exchanges a signed identity assertion for a
bearer token
curl -X POST https://app.speakeasy.com/oauth/workload/device-
agent/<organization_id>/token \
  -d 'grant_type=urn:ietf:params:oauth:grant-type:jwt-
bearer&assertion=<signed_jwt>'

# Step 4: Use the bearer token to ingest hooks, naming the
project
curl -X POST https://app.speakeasy.com/hooks/ingest \
  -H 'Authorization: Bearer <token>' \
  -H 'Gram-Project: <project_id>' \
  -d '<payload>'
```

— Full token exchange flow with endpoint, headers, scopes, and runnable example Detailed field behaviour but no direct call for a user to make

`server@2.27.0`

02 **Bulk directory role setup** NEW 83

Adds `previewDirectoryRoleSetup` and `applyDirectoryRoleSetup` for bulk directory role mapping setup; preview proposes a role per group and attribute value, and apply creates roles and replaces source mappings in one transaction.

Bulk-preview and apply directory role mappings for all groups and attribute values in one transaction.

```
$ # Preview proposed role assignments before committing
previewDirectoryRoleSetup({ organization_id: '<org_id>' })

# Review the proposal, then apply — creates new roles and
replaces source mappings atomically
applyDirectoryRoleSetup({ organization_id: '<org_id>',
grant_set: '<grant_set_id>' })
```

— Runnable preview/apply example with transactional semantics `server@2.27.0`

03 **Directory review queue and notifications** UPDATED INTEGRATION NEW

80

Adds `listDirectoryReviewItems` and `resolveDirectoryReviewItem` to surface and decide directory changes needing an admin decision — approving a new source maps it to chosen roles, approving a missing source removes its mappings, and every decision is recorded in the audit log — with Platform MCP exposing the same via `list_directory_review_items` and `resolve_directory_review_item` tools requiring explicit confirmation; also adds `getDirectoryReviewSettings` and `updateDirectoryReviewSettings` to control per-organization admin email notifications for pending directory review items (weekly reminders, up to four), which are unaffected by the `directory_review.opened_v1` and `directory_review.resolved_v1` webhooks.

— Mapping/removal mechanism and audit logging explained, no UI path given Concrete limits and webhook names but no example call

`server@2.27.0`

---

04 **MCP routes moved from /x/mcp to /mcp** BREAKING

80

Every `/x/mcp/*` route — including its OAuth endpoints, `.well-known` metadata, `/x/mcp/idp_callback`, and `/x/mcp/remote_login_callback` — now returns 404; MCP servers are served at `/mcp/{slug}`. Login flows in progress on `/x/mcp` at deploy time and shared-authorization-server sessions bound to an `/x/mcp` resource must re-authorize on `/mcp`.

— Clear migration note naming every affected route `server@2.27.0`

---

05 **Tunnel dashboard and server metrics** NEW

68

gram tunnel 0.2.0 adds a dashboard showing connected agents, network checks, and MCP request history, backed by the new `GET /rpc/tunneledMcp.getServerMetrics` endpoint; agents report aggregate HTTP progress to a gateway with diagnostics enabled without collecting payloads or making MCP calls.

— Dashboard and diagnostic mechanism described, no UI path given `tunnel@0.2.0`

---

06 **Platform MCP redeploy_deployment tool** NEW

63

Adds Platform MCP `redeploy_deployment` tool for previewing and confirming a whole-project deployment retry or rollback, using idempotency receipts so retries reuse the same deployment.

— Names the tool and idempotency mechanism, no example call `server@2.27.0`

---

07 **Per-server upstream OAuth clients** IMPROVED

60

Each MCP server and gateway member can now use its own upstream OAuth client instead of a provider-shared client; consent screens show each server only its own

clients, and existing connections made before a server was configured are asked to reconnect.

— Behaviour change explained but no config surface named `server@2.27.0`

08 **Directory people listing endpoint** NEW 60

Adds `listDirectoryPeople` session-only read that returns every live directory profile with its groups, membership status, account status, and sign-in history.

— Named endpoint with response fields, no usage example `server@2.27.0`

THINNER COVERAGE BELOW

09 **Org Gateway requires explicit opt-in** BREAKING 48

Org Gateway no longer becomes available automatically; an administrator must now explicitly opt in from the Organization Gateway card on the MCP Servers page.

— UI path given but no detail on prior default behaviour's impact `server@2.27.0`

└──▷ BREAKING ON UPGRADE

! Every `/x/mcp/*` route — including its OAuth endpoints, `.well-known` metadata, `/x/mcp/idp_callback`, and `/x/mcp/remote_login_callback` — now returns 404; MCP servers are served at `/mcp/{slug}`. Login flows in progress on `/x/mcp` at deploy and shared-authorization-server sessions bound to an `/x/mcp` resource must re-authorize on `/mcp`.

! Org Gateway no longer becomes available automatically; an administrator must now explicitly opt in from the Organization Gateway card on the MCP Servers page.

`dashboard@0.141.0` NOTES ↗ STABLE

gram dashboard adds directory membership visibility, bulk SSO role mapping, MCP server access lists, tunnel agent monitoring, and a unified MCP server list.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch dashboard@0.141.0
https://github.com/speakeasy-api/gram.git
# already have the repo? check out this version:
$ git checkout dashboard@0.141.0
```

Identities lists directory people who aren't members yet, with a Directory column and filters for membership status.

A new switch on the SSO page controls admin emails about directory changes for the organization.

A directory-changes review panel on the SSO tab surfaces new or removed groups and

attribute values affecting role mappings.

`Set up in bulk` on the SSO tab lets admins configure directory role mappings for every unmapped group and attribute value at once.

An MCP server's access list shows the directory groups and attribute values that grant each role.

+5 more

└─▷ BREAKING ON UPGRADE

- ! Stable device agent installer links move from `/v1/install/` to `/rpc/install/`; update any scripts or MDM configs referencing the old paths.
- ! MCP server headers can no longer use the caller's `Authorization` header as their source; existing headers of this kind are refused or dropped.
- ! The `hosted-mcp-server-key-writes` rollout flag is removed.
- ! Org Gateway no longer activates automatically; an administrator must opt in.

WAS THIS USEFUL? 👍 👎

api-mcp-server

Feedback ⓘ v3.7.0 NOTES ↗ RANK

> This window added a broad batch of new MCP operations spanning hosting deployment and CDN control, Reach forms and profile management, Node.js/headless build reporting, agent readiness scanning, and WordPress AI content retrieval, alongside bundled agent provisioning skills and a search-visibility change hiding 24 legacy hosting operations.

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

01 **Reach forms create, update, duplicate, and tagging operations**

NEW 85

Adds six new `reach_forms` MCP operations: `reach_forms_create` to create a new form, `reach_forms_update` to update an existing form, `reach_forms_duplicate` to duplicate a form, `reach_forms_add-submitter-tags` to add tags to form submitters, `reach_forms_remove-submitter-tag` to remove a tag from form submitters, and `reach_forms_update-template` to update a form template. Also extends the input schema of the existing `reach_forms_list` operation.

— Names all six operations and the schema change; no usage example given. v3.4.1

02 **CDN enable, disable, purge, and status operations** NEW

80

Adds four hosting CDN operations to the MCP server: `hosting_cdn_enable` to enable CDN for a hosting site, `hosting_cdn_disable` to disable it, `hosting_cdn_purge-cache` to purge the CDN cache, and `hosting_cdn_status` to retrieve the current CDN status for a hosting site.

— Names all four operations but lacks a usage example. v3.2.2

03 **Search visibility hidden for 24 hosting operations** IMPROVED

80

Hides 24 operations from search results—including `hosting_import-wordpress-website`, `hosting_deploy-wordpress-plugin`, `hosting_deploy-wordpress-theme`, `hosting_deploy-js-application`, `hosting_deploy-static-website`, `hosting_list-js-deployments`, `hosting_nodejs_list-builds`, `hosting_nodejs_start-build`, and `hosting_nodejs_analyse-failed-build`, among others—while keeping them executable via `execute`.

— Names hidden operations and the execute fallback mechanism. v3.2.0

04 **Unified hosting_deploy operation for multi-target deployment**

NEW 75

Adds `hosting_deploy` operation to deploy a website, Node.js app, WordPress plugin, or WordPress theme, and `hosting_get-deployment` to track build and deployment status via MCP.

— Names operation and deploy targets, no invocation example. v3.2.0

05 **Bundled agent provisioning skill definitions** NEW 75

Bundles three agent provisioning skill files — `skills/headless/steps/03-provision/ecommerce/SKILL.md`, `skills/headless/steps/03-provision/webapp/SKILL.md`, and `skills/headless/steps/03-provision/wordpress/SKILL.md` — covering e-commerce, web app, and WordPress setup workflows.

— Names exact file paths for each bundled skill. v3.6.1

06 **Reach profile sending-domain and settings operations** NEW 70

Adds `reach_profiles_connect-sending-domain` to connect a sending domain to a Reach profile, `reach_profiles_delete-sending-domain` to delete a sending domain from a Reach profile, and `reach_profiles_settings` to retrieve Reach profile settings, covering the connect/delete sending domain and get/update profile settings functionality described.

— Three operations named explicitly; a fourth mentioned only generically. v2.13.0

07 **Node.js build settings and headless build report operations** NEW 70

Adds `hosting_nodejs_build-settings-from-repository` to fetch Node.js build settings from a repository, and `hosting_headless_submit-build-report` to submit headless build reports via the MCP server.

— Names both operations, no usage example. v2.15.0 v3.7.0

08 **WordPress AI content generation result retrieval** NEW 60

Adds `wordpress_ai-content_generation-result` operation to retrieve the result of an AI content generation job.

— Single named operation, minimal mechanism detail. v3.0.0

THINNER COVERAGE BELOW

09 **Agent readiness scanning operations** NEW 55

Adds agent readiness operations including `agent-readiness_scans_website` to retrieve a website scan result, alongside operations to list checks and run website scans.

— Only one operation explicitly named; others described generically. v2.17.0

WAS THIS USEFUL? 👍 👎

memwhale

Feedback ⓘ v0.22.0 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.

> MemoryWhale's biggest addition this window is pre-run recall warnings that surface saved lessons before a known-failing command runs, alongside a full Gemini CLI integration, tightened fix suggestions built on program-stem fingerprinting, and more resilient `mw integrate` / `mw doctor` coverage across Homebrew upgrades and additional agents.

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

01 **Pre-run recall warnings via PreToolUse hooks** NEW 95

Adds `mw integrate claude` and `mw integrate codex --capture` to install a `PreToolUse` hook in `~/.claude/settings.json` that warns agents with one line of context before a previously-failed command runs, powered by `mw-remember --before-hook claude|codex`, which matches failures by program stem across directory-scoped history and stays silent unless a saved lesson or confirmed fix exists. A new `MEMORYWHALE_HOOK_FEEDBACK=0` environment variable disables both pre-run warnings and post-run notes together, and lesson-first ordering promotes a match on program stem or error-line fragment (punctuation-insensitive) to the top of output, with a matching lesson alone sufficient to trigger a note even without fix evidence. `mw doctor`, the dashboard, and the MCP `stats` tool now report warnings shown, warnings followed by the program passing, and pre-run warnings that subsequently passed.

Wire up the pre-run recall hook for Claude Code so it sees a saved lesson before re-attempting a known-failing build.

```
$ mw integrate claude
```

Add the pre-run hook alongside the existing capture hook for Codex, then trust the changed hooks when Codex prompts.

```
$ mw integrate codex --capture
```

— Exact flags, hook path, and env var all named verbatim v0.22.0

02 **Gemini CLI integration with MCP and capture** NEW 95

Adds `mw integrate gemini` to install MemoryWhale MCP in `~/.gemini/settings.json`, and `mw integrate gemini --capture` to register an `AfterTool` hook for `run_shell_command`; both support `--check`, `--dry-run`, and `--revert`. Gemini captured commands are stored with working directory, output, and exit code, and Gemini failures now count in `recent_errors` and `mw context --last-error`. `mw doctor` gained a `last captured` line per agent showing recency, exit-code coverage, or `never` with next steps, plus a `recall notes` line indicating whether hook feedback is active. Adding Gemini changed `memorywhale-core` 0.10.0's `SUPPORTED_AGENTS` from `[&str; 6]` to `&[&str]`, and recall-note parsers must now also accept `possible fix:` in addition to `the fix was:`.

Connect Gemini CLI to MemoryWhale MCP so the agent can read and write local debugging memory.

```
$ mw integrate gemini
```

Enable command capture for Gemini CLI so failed shell commands are recorded with their exit codes and surfaced in future recall notes.

```
$ mw integrate gemini --capture
```

— Exact commands, hook, and doctor output all specified `v0.19.0`

03

Stable integration paths and Codex/Cursor doctor checks

IMPROVED

80

`mw integrate` now writes stable `PATH`-based helper names (`/opt/homebrew/bin/mw-remember`, `/opt/homebrew/bin/mw-mcp`) instead of versioned Cellar paths, so hooks survive Homebrew upgrades, and it updates an existing install in place when its old path is gone, is a versioned Homebrew path, or points at the same file — covering `claude`, `rho`, `codex`, `cursor`, and `<client> --capture` targets. `mw doctor` now also covers Codex and Cursor (both the MCP entry and capture hook) and reports an entry as stale when it points at a missing or no-longer-current program.

After upgrading via Homebrew, re-register all agent integrations to replace stale versioned paths with stable PATH references, then verify with `mw doctor`.

```
$ brew update && brew upgrade memorywhale
mw integrate claude
mw integrate codex
mw integrate cursor --capture
mw doctor
```

— Exact paths and doctor coverage named with upgrade example v0.18.1

04 **Verified-fix suggestions and stem-based failure matching**

75

BREAKING

Adds `program_stem`, `stem_of_line`, `stem_for_failure`, `read_only_program`, `salient_error_line`, and `register_sql_functions` to `memorywhale-cli`; `FixCandidate::trusted` now encodes a rule limiting suggested fixes to verified evidence. `mw context --last-error` and the MCP `similar_failures` tool fingerprint failures by program stem so matches work across differently worded scripts, and the lesson prompt look-back is limited to ten runs for the same stem in the same directory, quoting the command truncated to 80 characters. Opening a store migrates its schema from 14 to 15, recomputing every error fingerprint with the program stem, and `Failure` now carries a `stem` field instead of `command` and `argv_json`, breaking any code that reads those fields.

— Named internal functions and migration, but no runnable example v0.21.0

UPDATED INTEGRATION Delphin integration updated to recognise `confirmed fix:` form

└──▷ BREAKING ON UPGRADE

! Opening a store migrates it from schema 14 to 15: every error fingerprint is recomputed with the program stem.

! In `memorywhale-cli`, `Failure` now carries a `stem` field instead of `command` and `argv_json` — any code reading those fields will break.

! In `memorywhale-core` 0.10.0, `SUPPORTED_AGENTS` changes type from `[&str; 6]` to `&[&str]`; code that relies on the array length must be updated.

! Tools that parse recall notes for 'the fix was:' must also accept 'possible fix:'.

v0.20.0 NOTES ↗ STABLE

Adds cross-agent fix sharing with `confirmed fix` / `possible fix` labels, provenance tracking, and recall-effectiveness stats in `mw doctor`, dashboard, and MCP `stats`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.20.0 https://github.com/wuisabel-gif/MemWhale.git
# already have the repo? check out this version:
$ git checkout v0.20.0
```

Adds `recall.hints_shown`, `recall.followed`, and `recall.worked` fields to the

MCP `stats` tool so agents can query how often recall notes led to a passing run.

Extends `mw context --last-error` with a `Recall:` line showing the best ranked fix candidate alongside its confidence label and provenance.

Adds `recall_hints` tracking to `mw doctor` output, reporting counts of notes shown to agents, followed, and worked.

Introduces `confirmed fix` and `possible fix` confidence labels in recall notes based on observed episode counts.

Surfaces per-agent provenance in recall notes, naming the agent that last ran the fix.

+3 more

WAS THIS USEFUL? 👍 👎

novamira

Feedback ⓘ v1.12.12 NOTES ↗ RANK

MCP server that gives AI agents full access to WordPress through PHP execution and filesystem operations

> Novamira's biggest addition this window is Read Mode, a Novamira Pro feature that restricts AI agents to read-only access via a new WP-CLI command, admin UI page, status visibility, and per-skill availability controls. It also shipped a PHP compatibility check skill, broader OAuth REST API coverage for MCP connections, Gutenberg hand-edit detection, and Kinsta cache clearing on AI Abilities changes.

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

01 Read Mode for read-only AI agent access NEW

91

Read Mode is a Novamira Pro feature (requires Pro 1.12.0 or newer) that restricts an AI agent to read-only abilities — disabling Novamira Visual, locking the Features page, and limiting the Abilities Hub to read abilities only. It can be checked and toggled with the `wp novamira read-mode` CLI command or from a new Read Mode page under Novamira in the admin UI, and its status is now visible to AI agents and Novamira HQ. Individual skills can also be made available while Read Mode is on, enabled from the skill's page or in the skill file.

Check whether Read Mode is currently active on your site — useful before handing an AI agent access to confirm it cannot make changes.

```
$ wp novamira read-mode
```

— Names CLI command, UI page, Pro version and restrictions across three releases.

v1.12.11 v1.12.10 v1.12.9

THINNER COVERAGE BELOW

02 PHP compatibility check skill NEW VULNERABILITY CHECK NEW

50

A new skill lets AI agents assess whether a site's plugins, theme, and custom code are ready for a newer PHP version, and propose an update or code change for each problem found.

— Describes behavior but gives no command or config surface. v1.12.11

03 OAuth MCP connections cover full REST API IMPROVED

45

Expands OAuth connections for MCP to cover the site's entire REST API, using the full access already approved by the administrator.

— States scope change but no endpoint or flag detail given. v1.12.10

04 **Hand-edit detection in Gutenberg editing** NEW 45

AI agents editing a Gutenberg page can now detect hand edits made after the agent last read the page, stopping instead of overwriting those changes.

— Explains mechanism but lacks a config or command surface. v1.12.10

05 **Kinsta cache clearing on AI Abilities changes** NEW 43

Adds support for Kinsta sites to let Novamira clear cached sign-in responses when AI Abilities settings change.

— Names host and trigger but no further mechanism detail. v1.12.10

UPDATED SUPPORTED PLATFORM PHP version flagging — end-of-life/outdated PHP detection

v1.12.12 NOTES ↗ LATEST STABLE

AI agents gain a site integrity check skill (also in Read Mode) and flag outdated PHP versions to prompt hosting-provider updates.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.12.12 https://github.com/use-novamira/novamira.git
# already have the repo? check out this version:
$ git checkout v1.12.12
```

Adds a site integrity check skill usable by AI agents in both standard and Read Mode — agents report findings and confidence levels, and require approval before making any changes.

Flags outdated PHP versions to AI agents so they can recommend updating PHP through the hosting provider.

WAS THIS USEFUL? 👍 👎

mcp-taiwan-legal-db

Feedback ⓘ v1.8.3 NOTES ↗ RANK

台灣司法院判決 + 全國法規資料庫 MCP server · Query Taiwan legal data from any MCP AI agent

> mcp-taiwan-legal-db v1.8.3 disables captcha sources by default, a breaking change for setups relying on them.

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

01 **Captcha sources disabled by default** BREAKING 33

Captcha sources are now off by default, reducing noise for deployments that do not need them. Any working setup that relied on them being enabled will need to explicitly re-enable them after upgrading.

— Names the change and migration note but no flag/config key given v1.8.3

└──▷ **BREAKING ON UPGRADE**

! Captcha sources are now off by default; any working setup that relied on them being enabled will need to explicitly re-enable them after upgrading.

WAS THIS USEFUL? 👍 👎

kordoc

Feedback ⓘ v4.24.0 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 biggest push this window is document-quality linting — block-level lint, a municipal budget-arithmetic checker, and an AI-writing/structure/filler-phrase gate for official documents — alongside structure-aware HWPX patching, template-based generation, and expanded OCR model selection and line export.

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

01 Structure-aware HWPX patching IMPROVED 98

Adds `PatchOptions.structure` (default on), CLI `--no-structure`, and MCP `patch_document` `structure` field to control whether `patch` reflects paragraph additions and deletions in HWPX documents. A `parse --keep-layout-tables` → markdown edit → `patch` workflow writes new content into an HWPX template while preserving formatting, bullet symbols, paragraph styles, and run composition; new paragraphs are cloned from the nearest same-symbol source paragraph (□·○·-·※·1.·가.…), changing only text. Table cell line counts now reconcile on patch — excess lines are added as cell paragraphs and deleted lines removed, instead of being appended to the last cell paragraph — MCP tool descriptions were updated to clarify when to use `patch_document` vs `generate_document`, and `patch_document` results now label partially-applied items with '(부분 적용)' plus a text summary.

Extract an HWPX template's layout as markdown, edit only the text, then patch it back — preserving all paragraph formatting and bullet styles.

```
$ npx kordoc parse 견본.hwpx --keep-layout-tables -o 편집본.md
# ... edit 편집본.md ...
npx kordoc patch 견본.hwpx 편집본.md -o 결과.hwpx
```

Patch an HWPX without reflecting structural paragraph additions or deletions — useful when only in-place text replacement is wanted.

```
$ npx kordoc patch 견본.hwpx 편집본.md --no-structure -o 결과.hwpx
```

— Workflow, defaults, and before/after behavior fully detailed v4.22.15

02 OCR detection and recognition model selection NEW 96

`--ocr-det` (API `ocrDet`) selects the detection model: `"mobile"` (default), `"server"` (PP-OCRv5 server det, 88 MB), or a local file path (absolute path + `--ocr-det-sha256`). `--ocr-det-long-side` (API `ocrDetLongSide`) sets the detection input's long-side resolution (default 960, range 32–4096). `--ocr-rec` and `--ocr-rec-dict` (API `ocrRec`, `ocrRecDict`) supply local recognition model/dictionary files with SHA-256 verification (`--ocr-rec-sha256`, `--ocr-rec-dict-sha256`), or select the built-in `"handwriting"` preset — a ~15 MB Korean handwriting recognition model downloaded on first explicit use at a fixed revision and SHA-256, not included in the `kordoc models` default bundle, leaving the default OCR model unchanged. The MCP `parse_document` tool gained matching `ocr_rec`, `ocr_det`, and `ocr_det_long_side` parameters, and the Python SDK gained `--ocr-det` / `--ocr-det-long-side` for parity.

Parse a scanned PDF via CLI choosing the server-grade detector at a high resolution for dense documents.

```
$ npx kordoc 스캔본.pdf --ocr --ocr-rec handwriting --ocr-det server --ocr-det-long-side 2048 -o 출력.md
```

Run OCR on a scanned handwritten Korean form and convert it to Markdown in one step.

```
$ npx kordoc 수기서류.pdf --ocr --ocr-rec handwriting -o 수기서류.md
```

Override the default OCR detection model and its long-side resolution when processing a document with small or dense text.

```
$ npx kordoc 스캔본.pdf --ocr --ocr-det <model> --ocr-det-long-side <pixels> -o 출력.md
```

Use the higher-accuracy PP-OCRv5 server detection model at increased input resolution when default mobile-model accuracy is insufficient for dense or small text.

```
$ npx kordoc 스캔본.pdf --ocr --ocr-det server --ocr-det-long-side 1280 -o 결과.md
```

Run OCR with a locally pinned recognition model and dictionary in an air-gapped environment, enforcing SHA-256 integrity so a mismatched file is never silently loaded.

```
$ npx kordoc 스캔본.pdf --ocr --ocr-rec /opt/models/rec.onnx --ocr-rec-sha256 <sha256> --ocr-rec-dict /opt/models/dict.txt --ocr-rec-dict-sha256 <sha256> -o 결과.md
```

— Every flag, value, size, and SHA check named `v4.22.10` `v4.22.6` `python-v0.2.0` `v4.22.0`

---

03 **Block-level document linting** NEW 95

Adds the CLI subcommand `kordoc lint-blocks <file|->` for block-level linting with JSON input/output, returning findings keyed by block ID and UTF-16 position. The same capability is exposed via a `lint_blocks` request type on the resident worker (protocol 2, capability `lint.blocks`), a `blocks` parameter on the MCP tool `lint_document`, and KordocClient.lint_blocks() in the Python SDK; the protocol 2 per-line request size limit was raised from 1 MiB to 8 MiB to accommodate the text payloads this requires.

Lint a HWPX file block-by-block in CI, receiving findings with block ID and UTF-16 position for precise inline feedback.

```
$ npx kordoc lint-blocks 보고서.hwpx
```

Pipe a JSON block payload into the linter from stdin for integration with a live editor that assembles blocks programmatically.

```
$ cat blocks.json | npx kordoc lint-blocks -
```

— Full command, protocol detail, and size limits all given `v4.24.0`

---

04 **Budget-document arithmetic linting** NEW 95

`kordoc lint --budget` (backed by `src/hwpx/budget-lint.ts`) validates Korean municipal budget reports: arithmetic cross-checks (`단가 × 수량 × 기간 = 금액`) with thousand-won/million-won unit conversion, GFM table subtotal/total recalculation, foreign-currency notation (`$` / `USD` → '달러'), and baseline-period labelling, exiting 1 on violation. The same checks are exposed via the `budget` parameter on MCP tool `lint_document`, and the library now exports `lintBudgetText`.

Run budget-document validation on a municipal budget report and fail CI if arithmetic or table-total errors are found.

```
$ npx kordoc lint 예산서.hwpx --budget --gate --preset 보고서
```

— Exact formula, flag, exit behavior and MCP exposure specified v4.23.0

05 **Template-based HWPX generation** NEW 88

`kordoc generate --template <견본.hwpx>` generates a new HWPX document styled from a department template file, extracting title-table, summary box, section/reference bands, symbol paragraphs, and data-table formats as role-typed archetypes. The same capability is exposed via the `template_path` parameter on MCP tool `generate_document` and the library call markdownToHwpx({ template }) (via `template-scan.ts`, `gen-from-template.ts`, `template-zip.ts`).

Generate a new HWPX report that inherits a department's house style from an existing template file.

```
$ npx kordoc generate 본문.md --template 견본.hwpx -o 보고서.hwpx
```

— Flag, MCP param and library API all named v4.23.0

06 **Gongmun gate for AI-written or purposeless documents** NEW 88

`kordoc lint --gate [--preset 보고서]` (exits 1 on failure) and `kordoc generate --gate` (suppresses file output on failure) block official documents that reviewers would read as AI-generated or purposeless, implemented in `src/hwpx/gongmun-gate.ts`. A `gongmunGate` library export and the MCP `generate_document` `gate` field (returning only the violation list on failure) provide the same check programmatically.

Gate a generated report before writing the file — if the document reads as AI-written or lacks a clear purpose, no HWPX is produced and the violation list is printed instead.

```
$ npx kordoc generate 보고서.md --preset 보고서 --gate -o 보고서.hwpx
```

Run all three lint passes together on a draft to catch spelling, structural, and AI-style issues before submission.

```
$ npx kordoc lint 보고서.hwpx --gate --logic --slop --preset 보고서
```

— Exit behavior, file path, and MCP field specified v4.22.17

07 **OCR line export with bounding boxes and confidence** NEW 88

`--ocr-lines` and `--ocr-lines-all` CLI flags (API `ocrLines`) export per-page OCR line text, ink bounding boxes (raster px), and confidence scores for PDF and image

inputs; `--ocr-lines-all` also emits low-confidence dropped lines under a `dropped` key. The typed output `OcrPageLines` (with `OcrModelFile`, `OcrLine`, `OcrBox`) is exposed in the Python and Java SDKs, with the Python SDK gaining the `--ocr-lines` flag for parity.

Extract per-line OCR bounding boxes and confidence scores from a scanned PDF in JSON format for downstream NLP or layout analysis.

```
$ npx kordoc 스캔본.pdf --ocr --ocr-lines --format json
```

Extract per-line OCR bounding boxes and confidence scores from a scanned PDF — including lines that were dropped for low confidence — to feed a downstream layout or quality pipeline.

```
$ npx kordoc 스캔본.pdf --ocr --ocr-lines-all --format json -o 결과.json
```

— Flags, dropped-line key, and SDK types all named `python-v0.2.0` `v4.22.0`

08 **Inline math rendering in generated documents** NEW 88

`--inline-math` (MCP `inline_math`, config key `inlineMath`) on the `generate` command converts `$…$` expressions in body text and table cells into native Korean math objects using Pandoc delimiter rules; off by default, leaving existing output byte-identical when disabled.

Convert a Markdown report containing inline math expressions into an HWPX file with native Korean math objects instead of raw `$…$` literal text.

```
$ npx kordoc generate 보고서.md --preset 보고서 --inline-math -o 보고서.hwpx
```

— Flag, config key, and default behavior specified `v4.21.31`

09 **OCR extraction of images embedded in document body** NEW 88

Setting `ocr: true` or `"force"` (or supplying an `OcrProvider` function) extracts text from embedded images inside HWP/HWPX body text via the same OCR path as standalone image input, inserting results as paragraphs or table cells immediately after each image block (or inside the cell for images within table cells); decorative images smaller than 24×24 px and WMF/EMF formats are skipped.

Extract text from images embedded inside an HWP/HWPX document body — useful when scanned figures or stamps are embedded in a government document and you need their

text content alongside the surrounding prose.

```
$ npx kordoc 문서.hwpx --ocr -o 문서.md
```

— Trigger values, skip thresholds, and placement specified v4.21.23

10 **Dry-run preview for form filling** NEW 84

`fill --dry-run -j fields.json` previews, without writing any file, which cell each key or array index would fill, reporting key collisions via an `overwrittenBy` field.

Inspect which cells a fields file would populate before committing a batch fill — useful for catching collisions (`overwrittenBy`) in complex multi-header forms.

```
$ npx kordoc fill --template gian -j 값.json --dry-run
```

— Command and collision field both given v4.21.25

11 **Configurable line-fill safety ratio** NEW 76

Adds `autoFit.safety` to the document config, setting the upper bound on the usable-width ratio (default 0.98, range 0.8–1) so documents rendered outside Hangul — such as rhwp preview or PDF-based apps — can be tuned down to prevent overfull lines.

When generating reports for apps that render via rhwp preview or PDF, lower `autoFit.safety` below the 0.98 default to reduce paragraphs that overflow into a short final line.

```yaml
autoFit:
  safety: 0.96
```

— Default, range, and use case given with example v4.21.16

12 **Document-structure lint pass** NEW 73

`kordoc lint --logic` runs 9 `LOGIC_*` rules (implemented in `logic-lint.ts`) to verify that reports and plans follow a background → proposal → execution structure, with `docs/gongmun-logic.md` as the canonical structure reference, synchronized to `references/logic.md` inside the gongmun skill by `sync-meta` . Up to 5 logic warnings are attached to the CLI and MCP generation output channel across all applicable presets.

Run all three lint passes together on a draft to catch spelling, structural, and AI-style issues before submission.

```
$ npx kordoc lint 보고서.hwpx --gate --logic --slop --preset 보고서
```

— Rule count and file named but rules not enumerated v4.22.17

13 **AI-style filler phrase lint pass** NEW 73

`kordoc lint --slop` runs 11 `SLOP_*` rules (implemented in `slop-lint.ts`) to flag AI-style filler phrases, abstract outcome claims, and translation-style repetition. Up to 5 slop warnings are attached to the CLI and MCP generation output channel across all applicable presets.

Run all three lint passes together on a draft to catch spelling, structural, and AI-style issues before submission.

```
$ npx kordoc lint 보고서.hwpx --gate --logic --slop --preset 보고서
```

— Rule count and file named but rules not enumerated v4.22.17

14 **Cell coordinates for filled form fields** NEW 72

Adds a `filled[].target` field — `{ table, row, col }` — to `--keep-layout-tables --keep-empty-cols` extraction output, exposing the document-order table number and grid row/column for every filled cell, including nested tables.

— Field shape given but no usage example v4.21.25

15 **Supply-chain compliance tooling** NEW 70

Adds `npm run sbom` to generate a CycloneDX software bill of materials, `scripts/check-licenses.mjs` to gate releases against an allowed-list of distribution-dependency licenses, and `docs/security-review.md` as a public-institution security review artifact.

Generate a CycloneDX SBOM for a kordoc deployment to satisfy supply-chain or procurement requirements.

```
$ npm run sbom
```

— Three named scripts/docs but thin per-item detail v4.21.31

16 **Chromium sandbox enabled by default for PDF output** BREAKING 66

Enables the Chromium sandbox by default for print/PDF output, with opt-out via `KORDOC_CHROMIUM_NO_SANDBOX=1` (root Linux only).

— Default change and opt-out variable named v4.21.31

17 **JSON output for generate command** NEW 60

Adds `kordoc generate --json`, which writes a single-line JSON object `{output, preset, bytes, warnings}` to stdout for pipeline consumption.

Emit machine-readable generation metadata so a downstream script can log file size and surface warnings automatically.

```
$ npx kordoc generate 보고서.md --preset 보고서 --json -o 보고서.hwpx
```

— Single flag and exact JSON schema given v4.23.0

18 **MCP write-tool path restrictions** NEW 60

Restricts MCP write tools from writing to hidden paths (`.claude`, `.ssh`, `.mcp.json`) and agent instruction files (`CLAUDE.md`, `AGENTS.md`).

— Exact protected paths named, no override documented v4.21.31

THINNER COVERAGE BELOW

19 **Expanded HWPX/HWP5 render support** IMPROVED 56

Adds render support for character shading (`charPr shadeColor`, HWP5 shade color), multi-column layout inside cells, vertical-align text boxes, page borders, column dividers, page numbers, equation captions, footnote/endnote placement, and cell-level table page breaks.

— Long list of named features without usage detail v4.21.31

20 **Totals-row placement control in generated tables** NEW 55

Adds the `--total-row` CLI flag and `total_row` MCP parameter to control where the totals row is placed in generated tables (`top`, `bottom`, or `keep`); the library now exports `placeTotalRow`.

— Flag and values named but no example shown v4.23.0

21 **Footnote block IR with table support** NEW 54

Adds `footnoteBlocks` to the IR output, preserving the full block structure — including tables — inside footnotes and endnotes. Markdown output renders them as single-line HTML tables inside (주: ...), with HWPX and HWP5 producing the same structure.

— Internal structure described, no user-facing flag v4.21.31

22 **Reduced npm install footprint** IMPROVED 49

Reduces the default `npm install kordoc` footprint from 611 MB to 446 MB by replacing the `@huggingface/transformers` tokenizer dependency with `@huggingface/tokenizers` for formula OCR.

— Before/after size given but nothing to action v4.22.0

23 **Character position in notation findings** IMPROVED 38

Adds a `column` field (1-based, first character of the matched span) to notation findings so callers can pinpoint the exact character position within a paragraph.

— Bare field addition with no usage example v4.23.0

24 **Startup validation for worker options** IMPROVED 38

Validates non-serialisable option values (NaN, Infinity) and non-existent engine paths before launching the worker, rejecting them at startup rather than at runtime.

— Behavior described without a flag or example python-v0.2.0

WAS THIS USEFUL? 👍 👎

excel-mcp-server

Feedback ⓘ v2.2.2 NOTES ↗ RANK

Excel MCP server: let AI (Claude, Cursor, Codex) create and edit Excel files with charts, PivotTables, slicers and formulas. No Excel required.

> Excel MCP Server's v2.0.0 release added sixteen new tools spanning PivotTables, charts, slicers, sparklines, VBA module writing and full formula calculation behind a breaking API consistency overhaul, and the three releases that followed layered on CSV import/export, formula auditing (`trace_formula`), `goal_seek` , large-workbook performance work and configurable resource-limit flags.

└──▷ **WHAT SHIPPED · 18 FEATURES** most completely described first ? what's the number?

01 PivotTable creation and large-source performance NEW

90

Adds `create_pivot_table` tool with `show_as` , `number_format` , `field_settings` (including `show_items` , `sort` , `group_dates` , `group_numbers`), `calculated_fields` , `layout` , `subtotals` , and `values_in` , producing real PivotTables that Excel refreshes. PivotTable creation from large sources was then sped up (92s to 54s) and now uses half the memory (2.1GB to 1.0GB) via XML-text pivot cache serialization, with a new `--max-pivot-cells` flag raising the supported source size to 3 million cells.

Allow PivotTables over very large sheets (e.g. 134k rows × 13 cols) while keeping individual read/write calls bounded.

```
$ excel-mcp-server --max-pivot-cells 5000000 --max-cells 500000
--max-file-mb 200
```

— Tool options, performance numbers and flag all named with example. v2.1.0 v2.0.0

02 CSV import and export NEW

90

Adds `import_csv` to read `.csv` or `.tsv` files into a sheet (new or existing) at a named cell, with auto-delimiter detection (`comma` , `semicolon` , `tab` , `pipe`), multi-encoding support (UTF-8 with/without BOM, UTF-16, `cp1252`), typed field parsing (numbers, `TRUE` / `FALSE` , ISO dates), and an `as_text` option to keep everything as text. Adds `export_csv` to write a sheet range as a CSV file, outputting raw ISO-date values or `formatted` display values, with configurable delimiter and encoding including `utf-8-sig` (BOM for Excel). Adds formula-injection protection: imported fields starting with `=` , `+` , `-` , or `@` are stored as text with Excel's quote prefix; `escape_formulas` (on by default) prefixes such cells with `'` on export.

Import a semicolon-delimited CSV exported from a European system (decimal commas, dates as DD/MM/YYYY) into a new sheet, keeping leading-zero fields as text.

```
$ import_csv file="/data/report.csv" sheet="Q1" cell="A1"
delimiter="semicolon" decimal="decimal" date_order="DMY"
```

— Full import/export options and security behavior detailed with example. v2.2.0

03 **Selective tool loading via --tools** NEW 90

Adds `--tools AREAS` flag (env var `EXCEL_MCP_TOOLS`) to expose only the core tools plus chosen areas (`workbooks`, `csv`, `sheets`, `data`, `formulas`, `formatting`, `tables`, `charts`, `pivots`, `slicers`, `names`, `notes`, `images`, `vba`), reducing context size; an unknown area stops the server at startup.

Restrict the server to only CSV and formula tools to minimise context window usage when running in an automated pipeline.

```
$ excel-mcp-server --tools csv,formulas
```

— Exact flag, env var, all areas and failure mode given. v2.2.0

04 **VBA module writing** NEW 85

Adds `--allow-vba-write` flag (or `EXCEL_MCP_ALLOW_VBA_WRITE=1` env var) to enable `write_vba_module` and `delete_vba_module` tools for `.xlsm` / `.xltm` files, supporting standard, class, `ThisWorkbook` and sheet modules.

Enable VBA macro writing so an AI client can generate `.xlsm` automation without Excel installed.

```
$ uvx excel-mcp-server stdio --allow-vba-write
```

— Flag, env var, tools and module types with runnable example. v2.0.0

05 **v2.0.0 breaking API consistency overhaul** BREAKING 85

Standardizes parameter names across tools (one name per concept: `sheet`, `range`, `at`, `source`, `name`, with units in names), breaking 1.x clients. `set_table` replaces `create_table` (`style` and `striped_rows` moved into `options`; new tables default to `TableStyleMedium2`); `create_pivot_table` replaces `create_summary_table`. `create_chart` removes `show_legend`, `data_sheet`, `x_axis_title`, and `y_axis_title`; `options.legend` now takes `bottom`, `right`, `left`, `top`, or `none`; `at` is now optional; `create_chart` is marked destructive because `replace`

overwrites a chart. Chart rendering now matches current Excel defaults (gray text, light gridlines, no rounded corners, one color per series, gaps between clustered columns, scatter plots points, straight lines without markers unless `smooth` or `markers` is set, horizontal bars list rows top-down). `read_range` now returns `{range, values, next_range?}` (drops `sheet` and `truncated`; page while `next_range` present) and omits trailing empty cells/rows; `find_cells` returns `matches` as `{sheet: {cell: value}}`; `describe_workbook` returns only `sheets`, `defined_names`, `has_vba`; `list_workbooks` returns `{path: size_bytes}`; `describe_sheet` drops `chart_count` (use `charts`) and empty fields; midnight dates read as `2026-01-31`. Mutating tools now return a small object (`sheet`, `range`, `name`, optional `note`) instead of a sentence. `set_sheet_layout` takes `column_widths_chars` and `row_heights_pt` as objects (e.g. `{"A": 20}`, `{"1": 30}`), and `auto_filter` is an object with `range`, `filters`, `remove`. `add_conditional_format` rules that format cells now require `fill_color` or `font_color`, reject mismatched fields, and assign new rules the next free priority. `add_data_validation` limits `error_message` and `prompt` to 255 characters, and lists now require `options` or `source`.

— Every renamed parameter and schema change enumerated for migration. v2.0.0

06

Configurable resource limits for large workbooks NEW

85

Adds `--max-loaded-cells` flag and `EXCEL_MCP_MAX_LOADED_CELLS` environment variable to cap the maximum number of cells loaded from a workbook (default 3,000,000), refusing oversized workbooks before parsing. Adds `--max-cells` flag and `EXCEL_MCP_MAX_CELLS` environment variable to cap the number of cells any single call may read, write, or format, plus `EXCEL_MCP_MAX_FILE_MB` as the environment-variable counterpart to the existing `--max-file-mb` flag.

Lower the per-workbook cell cap when serving untrusted files to reduce memory exposure.

```
$ uvx excel-mcp-server --max-loaded-cells 500000
```

Set the cell limit via environment variable for containerised or systemd-managed deployments.

```
$ EXCEL_MCP_MAX_LOADED_CELLS=500000 uvx excel-mcp-server
```

Set cell and file-size limits via environment variables for a containerised deployment without changing the startup command.

```
$ EXCEL_MCP_MAX_PIVOT_CELLS=3000000 EXCEL_MCP_MAX_CELLS=250000
EXCEL_MCP_MAX_FILE_MB=100 excel-mcp-server
```

— All flags and env vars named with runnable examples. v2.2.2 v2.1.0

---

07 **Formula calculation in read_range** IMPROVED 80

Improves `read_range` by adding formula calculation in `values` mode — covering ~260 functions including `LET`, `SORT`, `FILTER`, `UNIQUE`, `SEQUENCE`, `SUBTOTAL`, `AGGREGATE`, `INDIRECT`, `CELL`, `INFO`, `HYPERLINK`, structured references, and spill references (`A1#`), with uncalculable formulas reported in `uncalculated`, and now accepts whole-column and whole-row spans such as `B:B` and `2:3`.

Calculate dynamic-array formulas in a large sheet without opening Excel — useful in headless CI pipelines.

```
$ uvx excel-mcp-server stdio
```

— Function coverage and span syntax named, example is generic. v2.0.0

---

08 **Formula auditing via trace_formula** NEW 80

Adds `trace_formula` for formula auditing: returns `precedents` and `dependents` (one level or all, across sheets, defined names, table and spill references), plus `evaluate` showing sub-expression steps innermost-first, capped at 200 ranges and 100 steps with a `more` count for overflow.

— Mechanism and caps named but no example call shown. v2.2.0

---

09 **Goal Seek tool** NEW 70

Adds `goal_seek` to change a constant cell until a formula reaches a target value (tolerance 0.001, max 100 iterations), refusing invalid inputs such as formula-holding input cells or formulas the calculator cannot evaluate.

— Tolerance, iteration cap and validation named, no example. v2.2.0

---

10 **Faster large-workbook writes and scans** IMPROVED 70

Large-workbook edits are now dramatically faster: a 10-cell write to a 200,000-row sheet drops from ~50s to ~3s by parsing only the 128-row blocks touched and streaming the rest, with `describe_workbook` scanning the used range in raw XML text instead of parsing every cell (200,000 rows: 12s to 1.4s); this fast-edit streaming now also covers sheets with shared (fill-down) formulas, covering most large workbooks produced by Excel. `describe_sheet` on large sheets no longer parses every cell — it now reads notes, links, hidden rows, and used range from targeted XML rows instead, improving performance on oversized worksheets.

— Concrete before/after benchmarks and mechanism, no new command. Mechanism described but no measured numbers given.

v2.2.0 v2.2.2

---

11 **Chart creation with full type and styling support** NEW

65

Adds `create_chart` tool supporting all Excel chart types including `doughnut`, `radar`, `bubble`, and Excel 2016 types `waterfall`, `histogram`, `pareto`, `box_whisker`, `treemap`, `sunburst`, and `funnel`, with combo charts, secondary axes, trendlines, error bars, axis scale/ `log` / `reverse`, data labels, series colors/markers, and `scatter_style`.

— Extensive chart types and options named, no usage example. v2.0.0

12 **Expanded options for write_range, format_range, clear_range and copy_range**

NEW

65

Adds `write_range` support for `links` (hyperlinks to `http`, `https`, `mailto`, or intra-workbook targets with tooltips), `format_range` support for `locked` and `formula_hidden`, `clear_range` support for `clear: "rules"` to remove conditional formats and validation, and `copy_range` options `paste` (`all`, `values`, `formulas`, `formats`), `transpose`, and `skip_blanks`.

— Four tools' new parameters named without a runnable example. v2.0.0

13 **Native table creation via set_table** NEW

60

Adds `set_table` tool with `options` (style, header row, totals row with per-column functions, banded rows/columns, filter buttons), calculated columns, and table resizing via `range`.

— Options named in detail but no runnable example. v2.0.0

THINNER COVERAGE BELOW

14 **Pillow 13.0 requirement for image insertion** BREAKING

55

`insert_image` now requires Pillow 13.0 or later (previously 12.x only) for updated memory-safety and denial-of-service protections in its image decoders; Pillow 12.x is no longer accepted.

— Exact version requirement and reason given, simple migration. v2.2.2

15 **Slicer and timeline support** NEW

50

Adds `add_slicer` and `delete_slicer` tools supporting table and PivotTable slicers, timelines for PivotTable date fields, style, columns, sort, and shared slicer caches across multiple PivotTables.

— Tool and options named but no usage example given. v2.0.0

16 **Sparkline support** NEW

50

Adds `add_sparklines` and `delete_sparklines` tools for line, column, and win/loss sparklines with colors, emphasized points, axis options, and date axes.

— Tool and sparkline types named, no example provided. v2.0.0

17 **Range transform tool** NEW 45

Adds `transform_range` tool supporting `remove_duplicates`, `text_to_columns`, and `fill` (Fill Down, Right, and Series).

— Names the three transform modes without a usage example. v2.0.0

18 **New basic editing and cleanup tools** NEW 45

Adds `sort_range`, `replace_cells`, `set_defined_name`, `delete_defined_name`, `set_note`, `delete_note`, `insert_image`, `delete_image`, `delete_pivot_table`, `delete_chart`, and `set_workbook_settings` tools.

— Eleven tools named with no further mechanism described. v2.0.0

▷ BREAKING ON UPGRADE

- ! Pillow 12.x is no longer accepted; `insert_image` now requires Pillow 13.0 or later.
- ! Tool parameters and results are now consistent with one name per concept (`sheet`, `range`, `at`, `source`, `name`, with units in names) — 1.x clients break on upgrade.
- ! `set_table` replaces `create_table` (`style` and `striped_rows` moved into `options`; new tables default to `TableStyleMedium2`); `create_pivot_table` replaces `create_summary_table`.
- ! `create_chart` removes `show_legend`, `data_sheet`, `x_axis_title`, and `y_axis_title`; `options.legend` now takes `bottom`, `right`, `left`, `top`, or `none`; `at` is now optional; `create_chart` is marked destructive because `replace` overwrites a chart.
- ! Chart rendering now matches current Excel defaults: gray text, light gridlines, no rounded corners, one color per series, gaps between clustered columns, scatter charts plot points, line charts are straight without markers unless `smooth` or `markers` is set, horizontal bar charts list rows top-down.
- ! `read_range` returns `{range, values, next_range?}` (drops `sheet` and `truncated`; page while `next_range` is present) and omits trailing empty cells and rows. `find_cells` returns `matches` as `{sheet: {cell: value}}`. `describe_workbook` returns `sheets`, `defined_names`, and `has_vba` only (drops path, size, counts). `list_workbooks` returns `{path: size_bytes}`. `describe_sheet` drops `chart_count` (use `charts`) and empty fields. Midnight dates read as `2026-01-31`.
- ! Tools that mutate a workbook now return a small object (`sheet`, `range`, `name`, optional `note`) instead of a sentence; only data-returning tools keep an output schema.
- ! `set_sheet_layout` now takes `column_widths_chars` and `row_heights_pt` as objects (`{"A": 20}`, `{"1": 30}`), and `auto_filter` is an object with `range`, `filters`, and `remove`.
- ! `add_conditional_format` rules that format cells now require `fill_color` or `font_color`; fields that do not fit the rule type are rejected; new rules take the next free priority.

! `add_data_validation` `error_message` and `prompt` are limited to 255 characters; a list now requires `options` or `source`.

WAS THIS USEFUL? 👍 👎

fablecut

Feedback ⓘ v1.16.0 NOTES ↗ RANK

Zero-dependency browser video editor that AI agents can drive — JSON timeline, MCP + REST, live-reloading UI

> FableCut's four releases this window add full multi-project and version management, multi-sequence timelines, an extensive agent-drivable motion-graphics toolkit (compositions, shape layers, motion paths, masks, track mattes, motion tracking), delivery presets with batch export and loudness normalization, and major performance gains from a WebCodecs export pipeline and GPU-accelerated preview effects.

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

01 **Frame inspection API for agents** NEW 93

Adds `fablecut_frame` (MCP) and `GET /api/frame` to return a source frame, contact sheet of evenly spaced frames, or composed timeline frame as a JPEG (with source name, timestamps and clips); `text:true` returns luma levels (0–1), dominant colours, texture, saturation and per-cell motion statistics for vision-less agents; reading a local file by `path=` requires the `X-FableCut-Client` header.

Let a vision-less agent sample frames across a clip to measure luma, colour and motion without receiving image data.

```
$ curl -s 'http://localhost:3000/api/frame?text=true'
```

— Endpoint, header requirement, and runnable curl example. v1.15.0

02 **Clip and grade-layer masking** NEW 93

Adds `props.masks` to video, image, SVG, text, and adjustment clips supporting up to 8 masks per clip, with types `rectangle`, `ellipse`, `bezier`, and `freehand`, each configurable with `mode` (add, subtract, intersect, difference), `invert`, `opacity`, `feather`, `expansion`, and full keyframe support including on bezier outlines; agent patch ops `setMask`, `removeMask`, and `setMaskKeys` control masks programmatically, validated on `addClip` / `updateClip`, with `fablecut_scopes {mask:{clip}}` reporting mask coverage per clip. Also adds `shape:"bezier"` / `"freehand"` to `setGradeLayer` so color-grade layers support bezier and free-hand masks via the Color tab's Draw control.

Add a feathered ellipse mask to a clip to vignette-focus the viewer's eye, then invert it to darken the edges.

```json
{
  "op": "setMask",
  "clipId": "clip_07",
  "mask": {
    "type": "ellipse",
    "mode": "subtract",
    "invert": true,
    "feather": 40,
    "opacity": 0.85,
    "expansion": -10
  }
}
```

— Every mask type, op and limit named with example. v1.13.0

03 **Multi-project management via REST and MCP** NEW 90

Adds `GET /api/projects` and `POST /api/projects` REST endpoints and the `fablecut_projects` MCP tool for creating, listing, duplicating, renaming, tagging, archiving, and importing projects, each with a `./project.json` as the open document; a new Projects dashboard lets users create projects from Landscape, Vertical, Square, Portrait, 4K, and 720p canvases or saved templates.

List all projects and create a new 4K project via the REST API, useful for scripted project setup in a CI or agent pipeline.

```
$ # List projects
curl http://localhost:PORT/api/projects

# Create a new project
curl -X POST http://localhost:PORT/api/projects \
  -H 'Content-Type: application/json' \
  -d '{"canvas": "4K"}'
```

— Full endpoint, MCP tool and UI path named with runnable example. v1.16.0

04 **Delivery presets and export dialog controls** NEW 90

Adds delivery presets (YouTube, YouTube Shorts, Instagram Reels, TikTok, X, LinkedIn, Master, Broadcast, Web, Archive/FFV1, Custom) to the Export dialog with controls for codec (H.264, HEVC, AV1, VP9, ProRes, DNxHR, XDCAM HD422, FFV1), encoder (NVENC, Quick Sync, AMF), CRF/VBR/CBR, pixel format (4:2:2, 4:4:4, 10-bit, lossless), container, audio, and a file-name template; presets are saved per-project under `library/delivery/`. Adds `preset` and `delivery` parameters to

`fablecut_export` ; `setProject {delivery}` pins a project-level default; `fablecut_encode_profiles` lists available presets and usable encoders.

— Every preset, codec and param named; no runnable example. v1.15.0

05 **Animation presets and motion-graphics templates** NEW

87

Adds `applyPreset` , `addTemplate` , `setTemplate` , `savePreset` and `saveTemplate` agent patch ops, `GET` / `POST /api/presets` and `GET` / `POST /api/templates` REST endpoints, 33 named animation presets (In, Out, At playhead, Loop) stored in `library/presets/` and applicable from the Inspector's Motion section, and Templates (lower thirds, logo animations, social layouts) stored in `library/templates/` , each placed as a comp whose text, colour, and logo fields are editable in the Inspector.

Fetch all saved animation presets from the REST API to populate a custom agent UI or pipeline.

```
$ curl -X GET http://localhost:3000/api/presets
```

— Endpoints, counts and storage paths with example call. v1.14.0

06 **Versioned project saves and restore** NEW

86

Adds `GET /api/versions` REST endpoint and the `fablecut_versions` MCP tool for named version saves, auto-save every 10 minutes, pre-agent-edit snapshots, version comparison (clips added/removed/changed, length, media, settings), restore, final-cut marking, and full backup download, surfaced in a new Versions panel.

Save a named version before a major agent-driven rewrite, then list available versions for comparison or restore.

```
$ # Save a named version
curl -X POST http://localhost:PORT/api/versions \
  -H 'Content-Type: application/json' \
  -d '{"name": "before-agent-rewrite", "note": "manual
checkpoint"}')

# List all versions
curl http://localhost:PORT/api/versions
```

— Endpoint, MCP tool, and runnable save example given. v1.16.0

07 **Multi-sequence timelines within one project** NEW

86

Adds patch ops `addSequence` , `inSequence` , `openSequence` , `updateSequence` , and `removeSequence` for agent-driven editing of multiple timelines (e.g. 16:9 and 9:16 cuts)

that share media and compositions within a single project; each sequence keeps its own canvas, frame rate, IN/OUT, markers, mix, and delivery settings.

Add a 9:16 vertical sequence to an existing project and edit it without switching the open timeline, using MCP patch ops.

```json
# Agent patch ops to add and edit a vertical sequence
{
  "op": "addSequence",
  "name": "vertical-cut",
  "canvas": "9:16"
}

# Edit the new sequence without opening it
{
  "op": "inSequence",
  "sequence": "vertical-cut",
  "ops": [ ... ]
}
```

— Named patch ops with a worked agent example. v1.16.0

08 **Batch export across multiple deliveries** NEW 86

Adds batch export via `project.outputs` (populated with `+ Typical set`); agents use `fablecut_export {outputs:true | [...]}` and `setProject {outputs}`; REST callers use `POST /api/export/begin {deliveries:[...]}` to run all outputs with one render pass per frame-rate/range (up to 8 outputs each).

Kick off a full delivery batch (master + YouTube 1080p + 9:16 reel + square + 720p preview) in one REST call so every output shares a single render pass.

```
$ curl -s -X POST http://localhost:3000/api/export/begin \
  -H 'Content-Type: application/json' \
  -d '{"deliveries": ["Master", "YouTube", "Instagram Reels",
"TikTok", "Web"]}'
```

— Endpoint with payload and runnable batch example. v1.15.0

09 **Vector shape layers** NEW 86

Adds `setShape` and `removeShape` agent patch ops, validating `props.shapes` on `addClip` / `updateClip`, and a `shape` clip kind drawing vector items in `props.shapes` — rectangles, ellipses, polygons, stars, lines, and bezier paths — each with fill, stroke, and `trimStart` / `trimEnd` / `trimOffset` trim paths.

Create a shape layer with a filled rectangle and an animated draw-on trim path for a lower-third wipe.

```json
// Via fablecut_patch_project setShape op
{
  "op": "setShape",
  "clipId": "shape-01",
  "shape": {
    "kind": "rectangle",
    "fill": { "color": "#1a73e8" },
    "stroke": { "width": 0 },
    "trimStart": 0,
    "trimEnd": 1,
    "trimOffset": 0
  }
}
```

— Clip kind, properties, and runnable patch example. v1.14.0

10 **Track matte compositing** NEW 86

Adds `props.matte` (values: `alpha`, `alpha-inverted`, `luma`, `luma-inverted`) and `props.matteTrack` for track matte compositing, letting a clip be cut by the picture on another track; the `setMatte` agent patch op gives programmatic control, and `fablecut_scopes` now lists each clip's matte and blend mode.

Apply a luma track matte so a title clip is revealed only through the bright areas of the video on the track above.

```json
{
  "op": "setMatte",
  "clipId": "clip_42",
  "props": {
    "matte": "luma",
    "matteTrack": "V2"
  }
}
```

— Matte values, patch op and runnable example given. v1.13.0

11 **Motion tracking for clips and masks** NEW 82

Adds motion tracking stored on clips via `props.tracks`, supporting point or region tracking (template matching, sub-pixel, optional scale and rotation); masks can back-track and key their position, scale, and rotation via ◀ Back / On ▶ buttons. Adds the

`fablecut_track` agent tool to track a point, region, or mask and `apply` the result to a mask or clip, plus the `removeTrack` patch op, with tracks validated on `addClip` / `updateClip`.

— Tool, properties and UI controls named, no example. v1.13.0

12 **Nested compositions for agent-driven motion graphics** NEW

80

Adds `addComp`, `updateComp`, `removeComp`, `precompose` and `inComp` agent patch ops for programmatic composition management, plus compositions (`project.comps`): nested timelines with their own size, length, tracks, and clips, nestable 4 deep, placed as a `comp` clip on the main timeline.

— All patch ops and nesting limit named, no example. v1.14.0

13 **Loudness normalization on export** NEW

78

Adds loudness normalization to delivery exports targeting −14 (streaming), −16 (podcast), −23 EBU R128, or −24 ATSC A/85 LUFS with a true-peak ceiling via ffmpeg two-pass `loudnorm`; every exported file is verified with ffprobe against size, frame rate, codec, audio, loudness, true peak and platform file-size limits, with the report shown in the export queue and in the `fablecut_export` reply.

— Exact LUFS targets and verification named, no command. v1.15.0

14 **Motion paths and text-on-path** NEW

78

Adds `setMotionPath` and `setTextPath` agent patch ops for motion-path and text-on-path control, a `textPath` clip property (line, arc, circle, wave, shape outline, or pen path) with keyframable `pathOffset` for text-on-a-path on title clips, and bezier handles (`pin` / `pout`) on position keys for motion-path curves, with per-frame dot preview on the monitor and Auto-orient to rotate the clip along the path.

— Named ops, properties and path types, no example. v1.14.0

15 **Motion blur and anchor-point controls** NEW

78

Adds `motionBlur` (shutter angle) and `motionBlurSamples` clip properties to average sub-frame samples of a moving clip in preview and export, and `anchorX` / `anchorY` properties on every visual clip to set the pivot point for position, rotation, and scale.

Add a motion-blur effect to a moving clip so fast pans look cinematic in both preview and export.

```json
// In your project JSON or via fablecut_patch_project
{
  "op": "updateClip",
  "id": "clip-42",
  "props": {
    "motionBlur": 180,
    "motionBlurSamples": 8
  }
}
```

— Exact properties and a runnable example given. v1.14.0

16 **Proxy media generation workflow** NEW 76

Adds `fablecut_proxy` (MCP; actions: make, check, delete) and `/api/proxy` endpoints with SSE event `proxy` to create H.264 proxies (540p or 720p, frame-accurate, short GOP) stored under `analysis/proxies/`; the compact view reports `proxy:540p`.

— Endpoints, storage path, and status format named. v1.15.0

17 **Media metadata and search in Project bin** NEW 73

Adds `GET /api/media-info` to expose per-file frame rate, codecs, audio channels, sample rate, bitrate and size (read via ffprobe) in the Project bin, with search by name, codec, resolution or `#tag`; adds `updateMedia`, `setCollection` and `removeCollection` patch ops for agents.

— Endpoint and patch ops named, no worked example. v1.15.0

18 **Null objects and clip parenting** NEW 68

Adds a `setParent` agent patch op validating `parent` on `addClip` / `updateClip`, a `motion-graphics` skill covering recipes, a `kind: "null"` clip kind — an invisible transform carrier — and a `parent` property on visual clips for hierarchy-based motion control.

— Clip kind and patch op named, no worked example. v1.14.0

19 **WebCodecs-based export pipeline** IMPROVED 65

Switches video export decoding from `<video>` seeking to WebCodecs, delivering ~5× faster export (95.7 s → 18.8 s on a 28 s 720p test project) with frame-exact accuracy.

— Concrete before/after numbers but no actionable command. v1.16.0

20 **GPU-accelerated chroma key and color effects** IMPROVED 65

GPU-accelerates (WebGL2) chroma key, temperature, tint and RGB split in preview and export, reducing a 720p keyed frame from ~113 ms to ~3.6 ms on an RTX 4090 and from ~123 ms to ~12.5 ms on Intel UHD; CPU fallback runs when WebGL2 is unavailable.

— Concrete benchmark numbers but no actionable step. v1.15.0

21 **Offline media relink and replace** NEW 63

Adds `relinkMedia` and `replaceMedia` patch ops and `POST /api/relink/candidates` to handle offline media; offline files surface in `fablecut_status` and the compact view.

— Endpoint and patch ops named, no example given. v1.15.0

THINNER COVERAGE BELOW

22 **Six new blend modes** NEW 55

Adds six blend modes — `color-burn`, `exclusion`, `hue`, `saturation`, `color`, and `luminosity` — bringing the total to 17; `blend`, `matte`, and `matteTrack` are all validated on `addClip` / `updateClip`.

— Names all six modes but no usage example. v1.13.0

23 **NVIDIA GPU export profiles** NEW 50

Adds `nvenc-h264` and `nvenc-hevc` export profiles for hardware-accelerated video export via ffmpeg NVENC.

— Names the two profiles but no usage example. v1.16.0

24 **Idle-tab CPU usage reduction** IMPROVED 50

Reduces editor CPU use during idle tabs from 83% to 2.2% of a core by slowing the render loop to 4 fps and suspending the audio graph 3 s after the last sound.

— Clear before/after numbers, no user action needed. v1.15.0

WAS THIS USEFUL?  

tabularis

Feedback ⓘ v0.29.0 NOTES ↗ RANK

Open-source desktop SQL workspace with 3 built-in database drivers and 24 shipped driver plugins, including SQL Server, DuckDB, ClickHouse and Redis. Built-in MCP server for Claude, Cursor and Devin, SQL notebooks and visual EXPLAIN.

> Across four releases Tabularis added MCP database targeting for its `list_tables`, `describe_table`, and `run_query` tools, expanded the plugin API with new DDL builders and per-plugin SQL dialects, and shipped ChatGPT sign-in, multi-database Postgres browsing, file import, and a vault-based connection-sharing format, alongside persistent Postgres transactions, configurable plugin call timeouts, JDBC connection import, and a long run of data grid, filter, and tab-management refinements.

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

01 **MCP database targeting and audit logging** NEW 85

Adds `list_tables`, `describe_table`, and `run_query` MCP tools the ability to pick the target database per call, and records the target database used in the AI activity audit log.

Use the MCP `run_query` tool against a specific database when your connection has access to multiple.

📍 In the MCP tool call for `run_query`, set the `database` parameter to the target database name — e.g. `{ "query": "SELECT * FROM orders LIMIT 10", "database": "analytics" }` — so the query runs against that database rather than the connection default.

— Names exact MCP tools and shows a runnable call example. v0.29.0

02 **Connection management enhancements** NEW 85

Adds support for browsing multiple PostgreSQL databases on a single connection via a database selector in the schema explorer plus a default-database chooser when setting up or editing a connection, adds a pooler mode field to `ConnectionParams` for connection pooler support, imports JDBC URLs, query credentials, and SSL modes in the Connections dialog—pasting a string such as `jdbc:postgresql://host:5432/mydb?user=alice&password=secret&sslmode=require` auto-populates host, port, database, credentials, and SSL mode—and enables reordering connections within a group by drag-and-drop in the connections panel.

Browse multiple PostgreSQL databases on one connection without creating separate connection entries.

📍 In the sidebar, open a PostgreSQL connection, then click the database selector at the top of the schema explorer to switch between databases on the same connection.

Paste a JDBC connection string to onboard a new database without manually filling every field.

📍 In Connections, choose `Import JDBC URL`, paste a string such as `jdbc:postgresql://host:5432/mydb?user=alice&password=secret&sslmode=require`, and Tabularis populates host, port, database, credentials, and SSL mode automatically.

— Explains browsing mechanism with concrete UI navigation example. Names the config field but no usage detail. Exact JDBC URL example shows auto-populated fields. Bare UI description, no mechanism detail.

v0.29.0 v0.28.0 v0.27.0

---

03 **Data grid cell, column, and visual enhancements** NEW

85

Adds a column statistics popover in the column header menu, bulk cell-selection editing from the context menu, a random UUID cell action, the ability to open a URL or email address directly from a cell, copy column names as lines and as a JSON array, copy selection as an HTML table, a setting for default sort direction on column header click, optional alternating row background (zebra stripes), a keyboard column-picker to jump directly to any column in the grid, and a `Filter by this value` option in the cell context menu for instant column filtering, later extended with comparison and contains operators.

Instantly filter a result set to rows matching a specific cell value without writing a WHERE clause.

📍 In the datagrid, right-click any cell › select `Filter by this value` to apply a column filter scoped to that value.

— Enumerates many named grid actions without exact navigation for each. Names exact context-menu action and operators with example.

v0.29.0 v0.28.0 v0.27.0

---

04 **Plugin and driver API extensions** NEW

80

Adds a `drop_table_sql` DDL builder and sends connection context to all DDL builders, lets drivers declare plugin-owned connection fields and secrets, lets drivers declare their SQL syntax so the editor and tooling adapt per-plugin dialect, allows file-based plugins to use in-memory databases, introduces optional driver-owned table query templates letting drivers supply custom query templates for table views, adds a

configurable plugin call timeout with a per-plugin override so long-running plugins can be tuned independently, and sends a cancel notification to a plugin when its call timeout is reached to enable cleaner plugin-side cleanup.

— Lists every named API surface but no usage example. Explains per-plugin override and cancel mechanism, no exact config key.

v0.29.0 v0.26.0

---

05 **Persistent Postgres tab transactions** NEW 80

Keeps a PostgreSQL tab's transaction alive across multiple query runs, so a transaction begun with `BEGIN;` in one run persists into the next within the same tab until an explicit `COMMIT;` or `ROLLBACK;`.

Keep a long-running Postgres transaction open across multiple editor runs in the same tab — useful for staging multi-step changes before committing.

> 📍 In a PostgreSQL connection tab, run `BEGIN;` then execute subsequent queries across multiple runs — the transaction remains alive between runs until you explicitly `COMMIT;` or `ROLLBACK;`.

— Exact SQL commands shown demonstrating transaction persistence. v0.26.0

---

06 **Tab management and display improvements** NEW 75

Adds `Ctrl/Cmd+Shift+T` to reopen the last closed editor tab, lets you duplicate an editor tab from the tab context menu, switches editor tabs in most-recently-used order, shows the schema name on table tabs when multiple same-name tables are open simultaneously, and names result tabs automatically from the leading SQL comment in each query, e.g. a query prefixed with `-- Customer Summary` produces a tab labelled Customer Summary.

Name a result tab descriptively so multi-query scripts are easy to navigate.

> 📍 In the editor, prefix your query with a leading SQL comment: -- Customer Summary SELECT * FROM customers; — the result tab will be labelled Customer Summary.

— Names shortcut and ordering behavior across combined tab actions. Exact mechanism and SQL comment syntax shown in example.

v0.28.0 v0.27.0

---

07 **Structured filter panel improvements** IMPROVED 70

Improves the structured filter panel with a Match all / Match any (AND/OR) toggle, text-friendly operators (contains, starts with, etc.) for string columns, a dropdown to pick values from the column's distinct values, and inline rendering instead of an overlay.

— Explains UI mechanism in detail but no runnable example. v0.28.0

---

08 **Auto-refresh table data** NEW 70

Adds auto-refresh of table data on a configurable interval in the datagrid, selectable from a toolbar interval picker.

Keep a live dashboard table current without manual refreshes by enabling auto-refresh on any table view.

📍 Open a table in the datagrid, locate the auto-refresh interval selector in the toolbar, and choose an interval (e.g. 5 s, 30 s, 1 min) — the grid will re-query on that schedule automatically.

— Clear mechanism with concrete interval options in example. v0.28.0

09 **Background notification for long-running queries** NEW 70

Notifies when a long-running query finishes in the background, with a configurable threshold (shown as 20 seconds in settings) before the alert fires.

Catch the moment a slow analytical query completes while you work in another tab — the background notification fires as soon as the result is ready.

📍 In the editor, run a long-running query, then switch to another tab. When the query finishes, a notification appears in the toolbar alerting you to return and inspect the result.

— Explains mechanism, threshold, and gives usage example. v0.28.0

10 **Convert selection to SQL list** NEW 60

Adds 'convert selection to SQL list' in the editor, turning a highlighted value set into an inline SQL list.

Convert a highlighted list of values in the editor into a SQL-ready IN-list without manual formatting.

📍 In the editor, select the values you want to convert, then use the 'convert selection to SQL list' action from the editor context menu.

— Exact editor action named with runnable usage example. v0.27.0

11 **Expanded command palette with contextual actions** IMPROVED 60

Expands the command palette with contextual actions surfaced from the current UI context, such as table-level actions (Copy, Filter, Export) appearing inline when relevant.

Surface a contextual action for the currently selected table or query result without leaving the keyboard.

> 📍 Press the command palette shortcut, then type the name of a table-level action (such as `Copy`, `Filter`, or `Export`) — contextual actions for the active UI element now appear inline.

— Explains contextual surfacing with example but no exact command. v0.26.0

THINNER COVERAGE BELOW

12 **ChatGPT sign-in and Requesty AI provider** NEW 45

Adds ChatGPT sign-in and plan usage, and Requesty, as new AI provider options in Settings.

— Names providers but describes only sign-in, no mechanism. v0.29.0

13 **File import into database** NEW 45

Adds a file import feature to load data from files directly into the database, available from the table explorer's Import action or the main menu Import panel.

Load data from a local file into the database using the new import feature.

> 📍 In the table explorer, right-click a table and choose `Import` (or open the Import panel from the main menu), then select your source file to load data directly.

— Thin description; only generic steps, no format specifics. v0.29.0

14 **Vault connection sharing format** NEW 45

Introduces a vault file format and three-way merge for team connection sharing.

— Mentions format and merge strategy but no detail on conflicts. v0.29.0

15 **Theme tokens for UI customization** IMPROVED 45

Routes every UI color, radius, and font through theme tokens, enabling fully token-driven theme customization.

— Describes token-driven theming but no concrete keys or values. v0.26.0

16 **View definition and DDL copy for views** NEW 40

Shows view definition and copies DDL for views from the explorer.

— Bare description, no mechanism or example given. v0.28.0

17 **Query history filtering** NEW 35

Filters query history by outcome and time range.

— Bare one-line description, no mechanism or example. v0.28.0

18 **JSON viewer links and color swatches** IMPROVED 35

Supports clickable links and color swatches in the JSON viewer.

— Thin one-line description, no mechanism specified. v0.28.0

19 **CI design and accessibility checks** NEW 30

Adds design and accessibility checks to CI, ensuring every UI change is validated for accessibility compliance.

— Process-only change, no user-facing mechanism or example. v0.26.0

└──▷ ALSO FROM THESE RELEASES




0:00 / 0:25


0:00 / 0:18


0:00 / 0:35


0:00 / 0:22





WAS THIS USEFUL? 👍 👎

mcp-grafana

Feedback ⓘ v2.1.0 NOTES ↗ RANK

MCP server for Grafana

> mcp-grafana v2.1.0 adds datasource-managed alert rule creation, renders interactive Prometheus charts directly in MCP hosts, declares per-tool RBAC permissions, ships signed releases with SBOMs, tightens timeouts and response-size limits, and makes `list_users_by_org` paginated (a breaking change).

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

01 Datasource-managed alert rules via Alertmanager tools NEW

95

`alerting_rules_write` and `alerting_routing_write` now create Mimir/Cortex or Loki ruler alert rules and Mimir/Cortex Alertmanager receivers with `managed_by: "datasource"`, targeting a datasource via `datasource_uid` or auto-selecting it when exactly one compatible datasource exists; existing rules and Alertmanager config are preserved.

Create a Loki datasource-managed alert rule group, targeting a specific datasource, without disturbing existing rules in that group.

```
$ alerting_rules_write(datasource_uid="abc123",
managed_by="datasource", ...)
```

— Names exact tools, fields and preservation behaviour with a runnable example v2.1.0

02 Paginated list_users_by_org (breaking) BREAKING

85

`list_users_by_org` is now paginated: it accepts new `query`, `limit` (default 100, max 1000), and `page` inputs and returns `{totalCount, page, perPage, orgUsers}` instead of a plain array of users. Any code that parses the tool output directly must now read `orgUsers` instead of the top-level array.

— Exact fields, defaults and migration note given v2.1.0

03 Timeout, cache and response-size limits hardened IMPROVED

80

`grafana_api_request` now times out after the Grafana client timeout (10s default, configurable with `--grafana-timeout`); `get_panel_image` render timeouts default to 60s and are capped at 120s; the client cache is bounded to 1000 entries per client type; Prometheus query and metadata responses larger than 10 MiB now fail with an error prompting you to narrow the query instead of being read without limit; and

`generate_deeplink` with `shorten: true` now returns an error containing the full link when a short URL cannot be created, instead of silently returning the long URL.

Raise the Grafana client timeout when managing slow Grafana instances, so `grafana_api_request` and other tools don't abort prematurely.

```
$ mcp-grafana --grafana-timeout=30s
```

— Names flags and exact limit values with a configuring example v2.1.0

04 **Interactive Prometheus charts via query_prometheus** NEW

60

`query_prometheus` now renders results as an interactive chart — time series, heatmap, bars, stat, or table, chosen automatically from the response shape — in MCP hosts that support MCP Apps, while plain text output is unchanged for other hosts.

— Describes the mechanism but gives no command to invoke it v2.1.0

THINNER COVERAGE BELOW

05 **Signed releases with SBOMs** NEW

55

Releases are now signed with keyless cosign, covering binary checksums, `.mcpb` bundles, and Docker images, each accompanied by SPDX SBOMs.

— Concrete artifacts named but no verification steps given v2.1.0

06 **Per-tool RBAC permission declarations** NEW

50

Every tool now declares its required Grafana RBAC actions in its `_meta` field under `grafana.com/requiredPermissions`, and the README's permissions column is auto-generated from those declarations.

— Names the metadata key but no steps to query it v2.1.0

└──▷ BREAKING ON UPGRADE

! `list_users_by_org` is now paginated: it accepts new `query`, `limit` (default 100, max 1000), and `page` inputs and returns `{totalCount, page, perPage, orgUsers}` instead of a plain array of users. Any code that parses the tool output directly must now read `orgUsers` instead of the top-level array.

WAS THIS USEFUL? 👍 👎

funplay-cocos-mcp

Feedback ⓘ v0.6.8 NOTES ↗ RANK

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 CodeBuddy IDE 4.0+ to its one-click MCP setup, introduces a read-only `check_asset_ready` asset-validation tool and a pixel-accurate `image-pixels` mouse input mode calibrated to screenshots, upgrades built-in Project Skills to v3 across all six supported clients, and tightens mouse/preview window targeting so unavailable windows now fail instead of silently redirecting.

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

01 **CodeBuddy IDE one-click MCP setup** NEW 80

Adds CodeBuddy IDE 4.0+ to the one-click MCP setup flow, writing server entries to the active Cocos project's `.mcp.json` and installing any missing Skills to `.codebuddy/skills`, using comment-preserving JSONC updates and project-specific server names so existing hand-edited configuration is left intact.

Bootstrap CodeBuddy IDE 4.0+ MCP integration for an active Cocos project without overwriting hand-edited comments or local Skills.

📍 Open your Cocos project in the Funplay MCP extension, then navigate to Configure + Skills to trigger one-click setup. The extension writes server entries to `.mcp.json` and installs any missing Skills to `.codebuddy/skills`, leaving your existing edits intact.

— Names config paths and preserves comments; has a walkthrough example. v0.6.8

02 **Pixel-accurate mouse input via image-pixels mode** NEW 80

Adds an opt-in `image-pixels` coordinate mode for mouse click/drag and preview input, bound to the captured window and project; calls are rejected if window geometry is stale or changed, and interrupted input reports uncertain delivery.

Use `image-pixels` coordinates from a calibrated screenshot to click a UI button at its exact pixel position inside the captured Game View.

📍 image-pixels

— Named mode with concrete binding behavior and usage example. v0.6.6

03 **check_asset_ready tool for asset readiness** NEW 65

Adds a read-only `check_asset_ready` tool in the Full profile that confirms an asset's imported metadata and stable UUID/URL identity via bounded native Asset Database queries.

Verify that a specific asset is fully imported and has a stable UUID before attempting to reference it at runtime.

```
📍 check_asset_ready
```

— Named tool with mechanism and a direct invocation example. v0.6.6

04 **Built-in Project Skills upgraded to v3** IMPROVED 60

Both built-in project Skills are upgraded to v3 with concise entrypoints and on-demand Cocos-specific references covering readiness, layout, nine-slice/importer settings, fonts/localization, and input validation.

— Lists covered topics but no usage path shown. v0.6.6

THINNER COVERAGE BELOW

05 **Screenshot calibration metadata fields** IMPROVED 55

MCP screenshot responses now include calibration metadata — `captureId`, actual PNG dimensions, viewport zoom/DPR, and crop origin — letting callers map captured pixels back to editor coordinates.

— Named fields but no standalone usage example. v0.6.6

06 **Unavailable window targets now fail input calls** BREAKING 55

Explicit unavailable window targets for mouse/preview input no longer fall back to another window — calls targeting an unavailable window now fail rather than silently redirecting.

— Clear before/after but no remediation steps given. v0.6.6

07 **Project Skills sync across six supported clients** IMPROVED 55

Project Skills version checks, diffs, backups, and restore now cover managed reference files across all six supported clients, preserving unrelated files and refusing to overwrite user-owned reference destinations.

— Describes safety guarantees but no command given. v0.6.6

└──▷ BREAKING ON UPGRADE

! Explicit unavailable window targets for mouse/preview input no longer fall back to another window — calls targeting an unavailable window now fail rather than silently redirecting.

WAS THIS USEFUL? 👍 👎

brightdata-mcp

Feedback ⓘ v4.1.0 NOTES ↗ RANK

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

> brightdata-mcp shipped a major v3.0.0 overhaul that replaces 46 web_data_* tools with a unified scraper workflow, alongside a new AI support assistant tool, a default-on tool policy change, prompt-injection protections for scraped content, and refined tool metadata and error handling.

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

01 Unified scraper workflow replaces web_data_* tools

BREAKING

90

46 `web_data_*` tools (covering Amazon, LinkedIn, Instagram, TikTok, YouTube, npm, PyPI and others) are removed and replaced by the `search_scrapers` → `get_scraper_details` → `run_scraper` workflow. `web_data_grok_ai_insights` and `web_data_reuter_news` are removed with no replacement, while `web_data_chatgpt_ai_insights` is renamed to `chatgpt_ai_insights` and `web_data_perplexity_ai_insights` is renamed to `perplexity_ai_insights`. The `bright-data-mcp`, `competitive-intel`, and `price-comparison` skills must be updated since older versions instruct the agent to call the removed tools.

— Full before/after with exact replacement workflow and tool names v3.0.0

02 All tools enabled by default; PRO_MODE removed

BREAKING

85

All tools — including `scrape_as_html`, `search_dataset`, `session_stats`, and all browser tools — are now on by default; use the `GROUPS` or `TOOLS` environment variables to restrict the server to a specific subset. `PRO_MODE` is removed and ignored — existing configs that set it continue to work without error, but the setting has no effect.

— Names exact env vars and affected tools for migration v3.1.0

03 Explicit read-only/destructive tool-hint annotations

NEW

80

Sets `readOnlyHint`, `destructiveHint`, and `openWorldHint` explicitly on every tool, marking `run_scraper` and AI insights tools as non-read-only, browser navigation and scrolling as non-destructive, and `scraping_browser_click_ref` / `scraping_browser_type_ref` as destructive — enabling clients to prompt for confirmation before destructive actions.

— Names specific hints and tools affected, enabling client behavior v4.1.0

04 **Discover tool removed** BREAKING — 70

The `discover` tool and its Discover API integration are removed from all tool groups and default configurations; replace usages with `search_engine` or `search_engine_batch` for source discovery, and `scrape_as_markdown` or `scrape_batch` for reading content.

– Clear removal with named replacement tools v4.0.0

05 **ask_brightdata_assistant support tool** NEW — 65

Adds the `ask_brightdata_assistant` tool to query Sophie, Bright Data's support assistant, about products, zones, errors, or setup; pass the returned `dialog_id` to continue a conversation thread.

– Names tool, assistant, and continuation mechanism via dialog_id v4.0.0

THINNER COVERAGE BELOW

06 **Sanitized and capped tool error output** IMPROVED — 55

Caps and sanitizes tool error output: token-redacted, stack-trace-stripped, and limited to 500 characters across all tools.

– Names concrete limit and redaction behavior, no direct reader action v4.1.0

07 **Scraper request tracing headers** NEW — 50

Adds `x-mcp-dataset-id` and `x-mcp-method` headers to scraper tool requests, enabling failed runs to be traced back to a specific scraper.

– Names headers but no reader-facing action described v3.0.0

08 **Prompt injection protection on scraped content** NEW — 50

Results from search, scrape, scraper, dataset, and browser tools are now marked as untrusted web content so the model treats returned text as data rather than instructions; error bodies from scraped sites receive the same marking.

– Describes mechanism but no direct reader action v3.1.0

09 **MCP routing instructions for tool selection** NEW — 50

Adds MCP `instructions` to the server so the model knows to prefer a pre-built scraper for known sites, then `scrape_as_markdown` / `scrape_batch`, and browser tools only for clicks, typing, or screenshots.

– Explains routing logic but is a model-facing, not user-facing, change v4.1.0

10 **full_page option on browser screenshot tool** IMPROVED — 40

Describes the `full_page` option on `scraping_browser_screenshot`.

— Names the exact option but gives no further mechanism v4.1.0

---

11 **Docker image runs on Node.js 24** IMPROVED 20

The Docker image now runs on Node.js 24.

— Bare version bump with no further detail v3.1.0

┗━▷ BREAKING ON UPGRADE

! The `discover` tool and its Discover API integration are removed from all tool groups and default configurations; replace usages with `search_engine` or `search_engine_batch` for source discovery, and `scrape_as_markdown` or `scrape_batch` for reading content.

! `PRO_MODE` is removed and ignored; existing configs that set it continue to work without error, but the setting has no effect.

! All browser tools, `scrape_as_html`, `search_dataset`, and `session_stats` are now enabled by default; operators who previously relied on them being off by default must explicitly restrict the active set via `GROUPS` or `TOOLS`.

! 46 `web_data_*` tools (Amazon, LinkedIn, Instagram, TikTok, YouTube, npm, PyPI and others) are removed; replace with the `search_scrapers` → `get_scraper_details` → `run_scraper` workflow.

! `web_data_grok_ai_insights` and `web_data_reuter_news` are removed with no replacement.

! `web_data_chatgpt_ai_insights` is renamed to `chatgpt_ai_insights` and `web_data_perplexity_ai_insights` is renamed to `perplexity_ai_insights`; any hardcoded references to the old names must be updated.

! The `bright-data-mcp`, `competitive-intel`, and `price-comparison` skills must be updated — older versions instruct the agent to call removed tools.

WAS THIS USEFUL? 👍 👎

knowl

Feedback ⓘ v5.28.0 NOTES ↗ RANK

Persistent memory for AI agents, for code, research and everyday life. Facts are typed, sourced, and retired when they change.

> knowl's biggest change this window is that skills now run without pre-approval by default, with `trust.autoApproveSkills` available to restore the old gate; it also adds domain ingestion profiles (`code` , `general` , `academic`) for `knowl_ingest` , broader global-store access across MCP tools, and a refactored AI provider registry.

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

01 **Domain ingestion profiles for knowl_ingest** NEW 95

Adds a `profile` parameter to `knowl_ingest` so each call selects a domain profile — `code` , `general` , or `academic` — with an unknown profile failing before any model call. `knowl config set ai.profile general` sets a per-store default profile (the global store defaults to `general` , persisted in `~/.knowl/config.json`), and custom profiles can be defined at `~/.knowl/profiles/<id>.yaml` , documented in `docs/domain-profiles.md` . The three built-in profiles drive distinct AI prompts for filter, extract, ask, and derive: `code` (unchanged default), `general` (people, preferences, plans, dates, how-tos), and `academic` (sources, claims, definitions, methods, results, people in the work).

Set a research store's default profile so every ingestion automatically applies academic extraction rules without per-call overrides.

```
$ knowl config set ai.profile academic
```

Define a custom domain profile — for example, a `legal` profile — so the AI ingestion prompts filter and extract domain-specific facts correctly.

```yaml
id: legal
filter: Extract only legal facts, case references, statutes,
obligations, and parties.
extract: Capture case names, statutory citations, defined terms,
obligations, and named parties.
ask: Is this text legal in nature?
derive: What legal obligations or rights does this text imply?
```

— Full mechanism, named parameters and file paths, with runnable examples. v5.26.0

02 **Skills run without pre-approval by default** BREAKING

85

Skills now execute without requiring `knowl skill approve <name>` to pin their bytes first — the default changed to auto-approve, so `knowl skill run <name>` works immediately in trusted repos. Repos with untrusted skills must explicitly run `knowl config set trust.autoApproveSkills false` to restore the old requirement that approval pin a skill's bytes before it can run.

Lock down skill execution in a repo containing third-party or untrusted skills so nothing runs until `knowl skill approve` has pinned its bytes.

```
$ knowl config set trust.autoApproveSkills false
```

Run a skill immediately in a trusted repo without a separate approval step.

```
$ knowl skill run <name>
```

— Clear before/after with runnable commands, but no deeper internals given. v5.28.0

03 **Global store access across MCP tools** IMPROVED

70

`knowl_update`, `knowl_ingest_atoms`, `knowl_decide`, `knowl_timeline`, `knowl_evidence_list`, `knowl_feedback`, `knowl_conflicts`, `knowl_context`, `knowl_synthesize`, `knowl_session_finish`, `knowl_gc_preview`, `knowl_gc_apply`, and `knowl_task_*` tools now operate against the global store when `knowl serve` is started outside a Knowl project directory. `knowl_update`, `knowl_evidence_list`, and `knowl_feedback` can also reach a global atom from inside a project, so IDs returned by search from the global store no longer fail with `not found`. `knowl_ingest` separately gained a `namespace: "global"` parameter, enabling writes to the global store from a session with no project open.

— Names every affected tool and parameter but gives no usage example. v5.28.0 v5.26.0

THINNER COVERAGE BELOW

04 **AI providers refactored into named registry** IMPROVED

55

AI providers (`openai`, `anthropic`, `ollama`, `custom`) are refactored into a named registry under `src/core/ai-providers/`; setting an unknown `ai.provider` now fails with the list of registered providers.

— Names the registry path and providers but no usage example given. v5.26.0

NEW INTEGRATION GitHub Copilot CLI session integration

NEW INTEGRATION Copilot AI provider gateway integration

└──▷ BREAKING ON UPGRADE

! Skills now run without `knowl skill approve` pinning their bytes first (default changed to auto-approve). Repos relying on the old approval gate must explicitly run `knowl config set trust.autoApproveSkills false` to restore it.

v5.27.0 NOTES ↗ STABLE

knowl v5.27.0 adds per-session reindex filtering (`--sessions`), per-profile Hermes init, and changes `approve --all` to skip `state` candidates by default (use `--include-state` to promote them).

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v5.27.0 https://github.com/dat999zx/knowl.git
# already have the repo? check out this version:
$ git checkout v5.27.0
```

Adds `--sessions <ids>` to `knowl reindex --transcripts` and `knowl transcripts extract` to limit a run to specific sessions, leaving the rest of the index untouched.

Adds `knowl init hermes` support for named Hermes profiles: writes the MCP entry and plugin into each `profiles/<name>/` directory that contains a `config.yaml` , not only the default home.

Adds `--include-state` flag to `knowl transcripts approve --all` to promote `state` candidates that `--all` now skips by default.

└──▷ BREAKING ON UPGRADE

! `knowl transcripts approve --all` no longer promotes `state` candidates — it skips them and prints a count; use `--include-state` to restore the previous behaviour.

WAS THIS USEFUL? 👍 👎

rulego

Feedback ⓘ v1.0.0 NOTES ↗

RuleGo is a lightweight, high-performance, embedded, next-generation component orchestration rule engine framework for Go.

RuleGo v1.0.0 delivers hop-budget loop protection, distributed locking primitives, cross-chain `ref://` references, multi-tenant RBAC with audit logs, rule-chain version snapshots, and hot config reload.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.0.0 https://github.com/rulego/rulego.git
# already have the repo? check out this version:
$ git checkout v1.0.0
```

UPDATED INTEGRATION Adds `msg_max_hops` configurable hop budget to the engine to prevent runaway message loops, with synchronous worker-pool fallback when the pool is saturated.

Adds `ref://` URI scheme inside rule chains to reference same-chain nodes and endpoints, with two-phase deploy and connection-pool cycle detection.

UPDATED INTEGRATION Adds distributed primitives `Locker`, `OnceGuard`, and `ActiveGuard` to the node pool — enabling schedule endpoints to fire exactly once across replicas and active/standby leader election; shared nodes now support alias-based lookup.

Adds `chainId:nodeId` cross-chain start-point syntax and `${}` dynamic `targetId` (resolving from msg/metadata/global/vars) to the flow/ref node, propagating `fromChainId` / `fromNodeId` for tracing.

Adds multi-tenant user management (roles, API Keys) and an admin-plane audit log with pluggable storage, rejected-request sampling, and MCP rule-chain change tool RBAC.

+17 more

NEW INTEGRATION BACnet/IP protocol collector

UPDATED INTEGRATION RabbitMQ configurable consumer queue and persistent publishing

UPDATED INTEGRATION Nacos connection-status reporting, soft-fail init, and log routing via WithLogger

NEW SUPPORTED PLATFORM BMP and WebP image decoder registration

UPDATED INTEGRATION streamsql stream-to-stream JOIN with WITHIN window and pivot columns input

└──▷ BREAKING ON UPGRADE

- ! The deprecated API surface `WithEndFunc`, `OnMsgWithEndFunc`, and `OnMsgWithOptions` are removed — callers must migrate to `WithOnEnd` and `OnMsg`.
- ! The `SubmitTack` method is removed with no replacement.
- ! The iterator node component is removed; rule chains must use the `for` node instead.
- ! The `utils/str` string template API (`NewTemplate`, `ExecuteTemplate`, `Template` interface) is removed; all template logic must migrate to `utils/el`.
- ! SSH authentication fields are merged: private key is now unified under `certKeyFile`; the separate `password` / `key` fields are removed and the two modes are mutually exclusive.
- ! rulego-server now requires Go 1.25 or later.

WAS THIS USEFUL? 👍 👎

mcpsnoop

Feedback ⓘ v0.28.0 NOTES ↗ RANK

Wireshark for MCP. A transparent proxy between your AI client and MCP server.

> mcpsnoop v0.28.0 extends session replay to CI and Streamable HTTP servers, adds multi-capture and outputSchema checks to its CI command, and can join distributed traces via OTLP headers.

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

01 **Session replay in CI and over Streamable HTTP** NEW 90

The `kerlenton/mcpsnoop` GitHub Action gains a `replay` input to drive a server with a recorded session and compare results against the recording, requiring no model or API key in CI, plus a `target` input for replaying a recording against a Streamable HTTP server endpoint. The `mcpsnoop replay` command itself now supports Streamable HTTP transport, so a recorded session can be replayed over either stdio or Streamable HTTP.

Replay a committed session against every CI build of your MCP server — no model, no API key, and the job fails if the server regresses.

yaml

```yaml
steps:
  - uses: kerlenton/mcpsnoop@v0.28.0
    with:
      replay: tests/mcp/claude-code-session.jsonl
      server: node dist/index.js
```

— Names exact Action inputs and a runnable CI replay example v0.28.0

THINNER COVERAGE BELOW

02 **Multi-capture and outputSchema checks in `mcpsnoop check`**

IMPROVED 55

`mcpsnoop check` now accepts several captures at once and can record agent runs in CI, and separately warns when a tool result violates its own `outputSchema`.

— Names command and outputSchema field but no exact flags v0.28.0

03 **Distributed trace joining via OTLP headers** NEW 55

mcpsnoop can join a caller's distributed trace when trace context arrives in HTTP headers, enabling end-to-end OTLP tracing across client, mcpsnoop, and server.

— Explains mechanism but no header name or config key given v0.28.0

WAS THIS USEFUL? 👍 👎

semble

Feedback ⓘ v0.7.0 NOTES ↗ RANK

Fast and Accurate Code Search for Agents. Uses 99% fewer tokens than grep+read

> semble v0.7.0 adds a new CLI command for inspecting indexed file types and speeds up indexing through parallelized chunking and batched embeddings.

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

01 **file-types CLI subcommand** NEW 75

Adds a `file-types` CLI subcommand to list which file extensions are indexed per content type, run as `semble file-types`, useful for checking indexing coverage before setting up `.sembleignore` rules.

Check which file extensions semble will index before setting up ignore rules with `.sembleignore`.

```
$ semble file-types
```

— Names exact runnable command and related config file v0.7.0

THINNER COVERAGE BELOW

02 **Faster indexing via parallel chunking and batched embeddings**

IMPROVED 40

Parallelizes chunking to speed up index builds across large codebases, and batches embeddings while upgrading model2vec for faster, more efficient indexing.

— Names mechanisms but no benchmarks or numbers v0.7.0

WAS THIS USEFUL? 👍 👎

anymd

Feedback ⓘ v8.5.4 NOTES ↗ RANK

Any file → clean Markdown for AI agents: PDF, Word, PowerPoint, Excel, EPUB, HTML and web pages, images (OCR), audio and video (metadata, subtitles, transcripts). A fast Rust MCP server and CLI that runs on your machine.

> anymd updated its Pro purchase flow so `anymd pro buy` now completes the purchase in-browser and automatically saves the license token locally.

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

01 In-browser Pro purchase with auto-saved license IMPROVED

67

`anymd pro buy` now opens `buy.sylphx.com/buy/anymd` in the browser and automatically saves the issued license token to the local machine, replacing the previous behaviour of just printing the Pro page. The `--no-browser` and `--json` flags are unchanged.

— Names exact command, URL and unchanged flags. v8.5.4

WAS THIS USEFUL? 👍 👎

reshapr

Feedback ⓘ 1.1.0 NOTES ↗ RANK

The open source runtime that turns your APIs AI-Native, without rebuilding them

> reshapr 1.1.0 introduces per-tool exposure control, new MCP programmatic discovery and execution endpoints, and expands Code Mode with audit logging and elicitation handling, alongside small editor UI improvements.

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

01 MCP tool discovery and code execution endpoints NEW

75

Implements `search_tools`, `get_api_types`, and `execute_code` endpoints in `McpController` for programmatic MCP tool discovery and code execution.

— Names concrete endpoints and controller, actionable for API consumers. 1.1.0

02 Per-tool exposure control via toolExposureMode NEW

65

Adds a `toolExposureMode` field that propagates across the entity, API, CLI, and proxy layers to control how individual tools are exposed.

— Names exact field and all propagation layers, but no value options given. 1.1.0

THINNER COVERAGE BELOW

03 Editor UI tool name and completion support IMPROVED

30

Adds tool name display on operation cards and completion support in the editor.

— Thin UI additions with no further mechanism described. 1.1.0

04 Code Mode audit logging and elicitation pre-flight NEW

25

Code Mode gains audit logging and an elicitation pre-flight strategy, though the mechanism behind either is not detailed beyond their addition.

— Named but undescribed; no mechanism or usage detail given. 1.1.0

WAS THIS USEFUL? 👍 👎

token-savior

Feedback ⓘ v4.24.0 NOTES ↗ RANK

3% plain. Structural code navigation + persistent memory engine.

> token-savior shipped a dense set of MCP-server capabilities this window: three new MCP prompts, a new `read_lines` tool, symbol-aware search and edit-context improvements, expanded discipline-guard enforcement, and a wide set of environment-variable controls for session indexing, sandboxing, and output size, alongside broader MCP SDK compatibility.

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

01 New read_lines tool NEW

95

Adds read_lines(file_path, start, end): 1-indexed, inclusive, numbered output by default (`numbers=false` for raw text); `end` is optional and omitting it returns a 60-line window, capped at 400 lines via `max_lines` ; also whitelisted for use inside `ts_execute` scripts.

Jump straight to a traceback line in a large file without knowing the enclosing symbol name — token-savior returns the 60-line window and names the enclosing function for you.

```
$ read_lines(file_path='src/runtime/cycle-reconciler.mjs',
start=312)
```

— Full signature, defaults, cap, and a runnable example v4.22.0

02 Project root resolution: TS_STICKY_ACTIVE and per-call routing

NEW

93

`TS_STICKY_ACTIVE=1` freezes `active_root` across multi-agent sessions so a parallel sub-agent cannot silently repoint the default project for others (explicit hints and `file_path` arguments still route correctly), while calls carrying an absolute `file_path` , `path` , or `target_file` now route to the nearest marker root that owns the path, registering linked worktrees on the fly without touching the shared `active_root` ; boot-signal resolution is ranked `CLAUDE_PROJECT_ROOT` → launch-directory root → `CLAUDE_PROJECT_DIR` , each validated for existence and a project marker before use.

Freeze the shared `active_root` in a multi-worktree session so a parallel sub-agent cannot silently repoint the default project for other agents.

```
$ TS_STICKY_ACTIVE=1 claude -p 'refactor auth module'
```

— Exact flag, behaviour, and a runnable command Detailed mechanism and resolution order, no direct command

v4.22.0

03 **SessionStart index tuning environment variables** NEW

85

Exposes `TS_SESSION_INDEX_MAX`, `TS_SESSION_INDEX_TYPES`, `TS_SESSION_INDEX_RECENT_DAYS`, `TS_SESSION_INDEX_EXCLUDE`, and `TS_SESSION_LINE_MAX` to tune the size and content of the SessionStart memory index. Processes sharing a project with different `TS_SESSION_INDEX_EXCLUDE` values — e.g. a daemon and a Claude Code session — will invalidate each other's index cache and trigger a full rebuild on every switch, so they should share the same value.

— Five named env vars plus a concrete cache-invalidation caveat v4.23.1

04 **TS_EXECUTE_MAX_CHARS output cap** NEW

84

`TS_EXECUTE_MAX_CHARS` caps `ts_execute` results at 60,000 characters by default, with cuts reported and logs/counters preserved — preventing oversized responses that clients like Claude Code would otherwise refuse.

Prevent `ts_execute` from returning a 150 k response that Claude Code refuses by capping output at a safe size.

```
$ TS_EXECUTE_MAX_CHARS=40000 claude -p 'summarise test coverage'
```

— Exact flag, default, and a runnable example v4.22.0

05 **Opt-in session discipline via TS_DISCIPLINE** NEW

83

`TS_DISCIPLINE=1` enables opt-in session instructions (977 characters) covering smallest-complete-change discipline, a reuse ladder via `search_codebase` with `semantic=true`, deletion-before-addition, and a closing skip summary.

Enable smallest-complete-change session discipline to keep an AI coding session focused on minimal edits and reuse before writing new code.

```
$ TS_DISCIPLINE=1 token-savior
```

— Exact flag, size, and mechanism with a runnable example v4.24.0

06 **Three MCP prompts: review, audit, lean** NEW

82

Adds three MCP prompts: `review` (change analysis including `get_change_impact` and `detect_breaking_changes`), `audit` (whole-project ranking as must/should/nice-to-have), and `lean` (task framed as smallest complete change). They surface as slash

commands in Claude Code and are free until called; `TS_PROMPTS=0` disables all three, useful in locked-down environments concerned about prompt injection via MCP prompts.

Remove the `review`, `audit`, and `lean` slash commands from Claude Code entirely — useful in locked-down environments where prompt injection via MCP prompts is a concern.

```
$ TS_PROMPTS=0 token-savior
```

— Names prompts, functions and toggle flag with a runnable disable command `v4.24.0`

07 **Sibling truncation in get_edit_context** IMPROVED 80

`get_edit_context` now returns only the 12 nearest siblings with real line numbers and a `siblings_total` field when the list is truncated, cutting response size by 29% (209,644 → 148,190 characters on a 40-symbol sample).

— Concrete before/after numbers and exact field name `v4.24.0`

08 **Auto profile manifest reduced** IMPROVED 80

Reduces the `auto` profile manifest from 18 to 13 always-on tools (~4,160 → ~3,220 tokens); hot slots reduced from 10 to 8 via `TS_AUTO_HOT_K`.

— Concrete before/after numbers and a named config flag `v4.22.0`

09 **Discipline guard enforcement expanded** IMPROVED 78

The discipline guard matcher now routes `get_function_source`, `get_class_source`, and `ts_execute` through the guard so reading a symbol counts as context; `TS_GUARD_READ_FREE_LINES` configures the line-count threshold above which native `Read` calls are refused by the guard.

— Named tools, flag, and mechanism grouped from the same guard system `v4.22.0` `v4.23.0`

10 **TS_LEVEL_BANDIT opt-in detail selection** IMPROVED 76

`TS_LEVEL_BANDIT=1` opts back in to Thompson-sampling detail-level selection for `get_function_source` and `get_class_source`; full source is now the unconditional default without the flag.

— Names flag and before/after default behaviour `v4.22.0`

11 **search_codebase, read_lines, and find_symbol refinements** IMPROVED 74

`search_codebase` hits now include an `in` field naming the containing symbol, resolving `who calls X` in one call instead of multiple Grep passes; `read_lines` results likewise report the containing symbol name when no single symbol covers the full requested range; search_codebase(max_results=0) results are now capped at 60,000

characters with a `_tronque` note; and `find_symbol` falls back to variables on a miss instead of returning `retry_with`.

— Names both fields and the problem each solves Two named, thin behaviour tweaks to core query tools

v4.23.0 v4.22.0

12 **Rationale field in get_edit_context** NEW 73

`get_edit_context` responses now include a `why` field returning the comment block (Python `#`, JSDoc `/** */`, `//`) glued directly above a definition as its rationale, capped at 8 lines; section banners are ignored.

— Names field, endpoint and exact cap but no direct command v4.24.0

13 **ts_execute interface enhancements** IMPROVED 73

`move_symbol` and `checkpoint` are callable again through `ts_execute`, restoring access to tools dropped from the `auto` profile's sandbox allowlist; `ts_execute`'s `script` argument also accepts `code`, `body`, and `js` as aliases.

— Named tools and aliases, grouped since each increment is thin v4.22.0 v4.23.1

14 **Architecture map in get_project_summary** NEW 72

`get_project_summary` now includes a two-line architecture map summarizing the most depended-on symbols (hubs) and largest graph communities, named by hub and production directory, generated with no model call (~400 characters per project).

— Specific mechanism and size, but no direct invocation shown v4.24.0

15 **MCP SDK version compatibility and auto-detection** IMPROVED 68

token-savior supports both MCP SDK generations via a real stdio handshake, tested against SDK 1.27 and 2.3 with options unknown to a given SDK silently dropped; it also auto-detects the installed SDK version and protocol `2026-07-28`, using 2.x callbacks when available and falling back to the 1.x path otherwise, pinned to `mcp>=1.25,<3`.

— Named versions and pin, but no reader-facing action v4.22.0 v4.24.0

16 **get_variable_source for top-level bindings** NEW 62

Adds `get_variable_source` to surface top-level `const` / `let` / `var` bindings with balanced brackets, available via `get_full_context` and `get_edit_context`.

— Names the tool and its integration points, no example given v4.22.0

17 **ES module indexing for .mjs/.cjs** NEW 62

Indexes `.mjs` and `.cjs` files so `get_functions` and `read_lines` cover ES modules; the index cache version was bumped to 4.

— Names file types, tools and cache version bump `v4.22.0`

└──▷ BREAKING ON UPGRADE

! Processes sharing a project with different `EXCLUDE_EXTRA` values (e.g. a daemon and a Claude Code session) will invalidate each other's index cache and trigger a full rebuild on every switch — give them the same value.

WAS THIS USEFUL?  

3dsmax-mcp

Feedback ⓘ v1.7.6 NOTES ↗ RANK

3ds Max MCP server

> 3dsmax-mcp v1.7.6 ships a hot-swappable C++ core, consolidates several commands into new unified calls, and delivers major performance gains for mesh and geometry QA tools, alongside a round of safety, stability, and installer improvements.

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

01 Consolidated tyFlow and property-inspection commands

BREAKING

90

Adds create_tyflow(preset=...) and connect_tyflow_operator(create_if_missing=...) as consolidated replacements for the removed `create_tyflow_preset` and `connect_tyflow_events` commands, and adds inspect_properties(target="modifier") as a consolidated replacement for the removed `inspect_modifier_properties` command.

Inspect modifier properties on an object using the new consolidated command instead of the removed `inspect_modifier_properties`.

```
$ inspect_properties(target="modifier")
```

Create a tyFlow with a preset using the new consolidated command instead of the removed `create_tyflow_preset`.

```
$ create_tyflow(preset="<preset_name>")
```

— Exact old and new command names and call syntax given v1.7.6

02 Mesh and geometry QA performance gains IMPROVED

78

`mesh_edit` repeated calls on one object now take a few milliseconds instead of ~400 ms, and `geometry_qa` is ~8x faster (72k triangles: 0.3 s, was 2.4 s, with safe mode enabled).

— Concrete before/after numbers for named tools v1.7.6

03 Server stability during client disconnects IMPROVED

70

3ds Max can now quit normally while an AI client is connected; unanswered calls return `MAX_BUSY` instead of freezing, and post-exit calls return `TARGET_UNAVAILABLE`.

— Named return states make behavior checkable v1.7.6

THINNER COVERAGE BELOW

04 **Shared-name safety in object targeting** IMPROVED 50

Tools that target one object now refuse a shared name and list candidates with their handles; batch tools skip and report shared names instead of picking arbitrarily.

— Describes behavior change but no specific tool names v1.7.6

05 **Hot-swappable native core** NEW 48

The bridge splits into a shell and a core so C++ functions can be added and tested without restarting 3ds Max.

— Mechanism described but no command or API named v1.7.6

06 **Nested material tree support in replace_material** IMPROVED 45

`replace_material` now also replaces materials inside Multi/Sub and other nested material trees.

— Names the tool and the nesting case it now handles v1.7.6

07 **Time-budgeted contact_check** IMPROVED 45

`contact_check` now uses a time budget and skips unmeasurable objects instead of failing.

— Names the tool and the failure mode fixed v1.7.6

08 **Group-member clone protection** IMPROVED 45

`clone_objects` now refuses cloning a member of a closed group instead of cloning the whole group.

— Names the tool and the exact guard added v1.7.6

09 **Windows installer bundling** IMPROVED 43

The Windows installer now bundles Python, dependencies, agent skills, and native bridges for 3ds Max 2023–2027.

— Names bundled components and supported version range v1.7.6

10 **Viewport rendering and curve tool additions** NEW 40

Adds support for Line objects in curve tools, V-Ray GPU for agent viewport V-Ray previews, and restores preview settings when stopped from the renderer UI.

— Three thin additions named but no mechanism given v1.7.6

11 **Removed commands and Tool Playground GUI** BREAKING 40

`refresh_plugin_manifest` and `list_tyflow_operator_types` are removed with no stated replacement, and the Tool Playground GUI is removed.

— Named but no replacement or migration path given v1.7.6

└──▷ **BREAKING ON UPGRADE**

! `create_tyflow_preset` is removed; callers must switch to create_tyflow(preset=...).

! `connect_tyflow_events` is removed; callers must switch to connect_tyflow_operator(create_if_missing=...).

! `inspect_modifier_properties` is removed; callers must switch to inspect_properties(target="modifier").

! `refresh_plugin_manifest` is removed with no stated replacement.

! `list_tyflow_operator_types` is removed with no stated replacement.

! The Tool Playground GUI is removed.

WAS THIS USEFUL? 👍 👎

pullmd

Feedback ⓘ v3.15.0 NOTES ↗ RANK

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 added new SITE-RECIPES.md actions for turning CSS-grid comparison layouts into real tables and replacing icon-only cells with readable text, a built-in recipe for finanzfluss.de comparison pages, a prepend-text action for ::before-only labels, and a breaking change that strips column-alignment padding from Markdown table output to shrink wide tables to roughly a quarter of their previous size.

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

01 grid-to-table recipe action for CSS-grid layouts NEW

95

Adds a `grid-to-table` recipe action in `SITE-RECIPES.md` that rebuilds CSS-grid comparison layouts (flat `<div>` cells with `grid-template-columns`) into real `<table>` elements, deriving the column count from repeat(N, ...), the track list, or an explicit `columns` setting; multiple grids found in the same container are merged into a single table.

Recover a structured Markdown table from a comparison page that uses a CSS grid instead of `<table>` markup, so LLM pipelines get rows and columns instead of a run of loose words.

yaml

```yaml
# In SITE-RECIPES.md, add a recipe for the target site:
# actions:
#   - action: grid-to-table
#     columns: 5
```

— Full mechanism plus a runnable SITE-RECIPES.md config example. v3.14.0

02 replace-with-text recipe action for icon-only cells NEW

86

Adds a `replace-with-text` recipe action that swaps matched elements with a fixed `text` value or with a named attribute's value via `attr`, so icon-only cells (check marks, `aria-label` icons, logo images) survive extraction as readable words, e.g. `attr: aria-label` in `SITE-RECIPES.md`.

Preserve icon-only cells (e.g. checkmark SVGs with an `aria-label`) as human-readable text in extracted Markdown tables.

```yaml
# In SITE-RECIPES.md:
# actions:
#   - action: replace-with-text
#     attr: aria-label
```

— Exact config fields and example config given. v3.14.0

03 **Markdown table output drops cell padding** BREAKING 80

Markdown table output no longer includes column-alignment padding, shrinking token-heavy wide tables substantially — a German Wikipedia city list drops from 123,508 to 83,804 characters and wide comparison tables shrink to roughly a quarter of their previous size. Cached results keep the old padded formatting until they are fetched again, so consumers relying on the padded format will see a layout change on next fetch.

— Concrete before/after numbers but no opt-out flag given. v3.14.0

04 **Built-in finanzfluss-vergleich recipe** NEW 73

Adds a built-in recipe `finanzfluss-vergleich` for finanzfluss.de `/vergleich/**` pages that outputs the provider comparison table as a Markdown table, rendering check-mark icons as `ja` / `nein` and deduplicating provider cards that appear in both mobile and desktop layouts.

— Named site pattern and behaviour but no manual invocation shown. v3.14.0

THINNER COVERAGE BELOW

05 **prepend-text recipe action** NEW 50

Adds a `prepend-text` recipe action that inserts a fixed string at the start of every matched element, enabling extraction of labels surfaced only via CSS `::before` pseudo-elements.

— No usage example or config snippet provided. v3.15.0

└──▷ BREAKING ON UPGRADE

! Markdown table output no longer includes column-alignment padding; cached results keep the old padded formatting until they are fetched again, so consumers that relied on the padded format will see a layout change on next fetch.

WAS THIS USEFUL? 👍 👎

agentcad

Feedback ⓘ v0.7.0 NOTES ↗ RANK

CAD CLI and MCP server for AI agents

> AgentCAD v0.7.0 bakes a required five-stage Inspect-Specify-Build-Validate-Deliver workflow into its agent skill and docs, adds dry-run and render flags to `agentcad run`, a summary flag for `agentcad inspect`, and configurable build-directory redirection for CI-friendly runs.

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

01 Configurable build directory via `build_dir` and `--build-dir`

NEW 90

Adds a `build_dir` key to `agentcad.toml` to redirect version directories, manifests, viewers, exports, and logs to a dedicated build directory, plus `agentcad init --build-dir <path>` and `agentcad run ... --build-dir <path>` flags so CI pipelines can use independent build roots without editing project config.

Redirect all build artifacts to a temp directory in CI so the source tree stays clean.

```
$ agentcad init --build-dir /tmp/cad-build && agentcad run model.py --label first --build-dir /tmp/cad-build
```

Keep build artifacts out of the source tree for every project run locally by setting a persistent build directory.

```toml
build_dir = "./build"
```

— Names exact config key, file, and CLI flags with runnable examples v0.7.0

02 Dry-run and render flags for `agentcad run` NEW 85

`agentcad run script.py --label test --dry-run` checks build metrics without consuming a version, and `agentcad run script.py --label first --render iso` generates an isometric render during a real build run.

Run a dry-run build to validate metrics and catch issues before committing a versioned output.

```
$ agentcad run script.py --label test --dry-run
```

— Exact flags and a runnable dry-run example given v0.7.0

03 **Summary flag for Inspect stage** NEW 65

`agentcad inspect FILE --summary` adds a concise topology and edit-risk summary output during the Inspect stage.

— Exact flag named but no runnable example provided v0.7.0

THINNER COVERAGE BELOW

04 **Five-stage Inspect-Specify-Build-Validate-Deliver workflow** NEW 50

Ships the required five-stage `INSPECT -> SPECIFY -> BUILD -> VALIDATE -> DELIVER` workflow into the installed agent skill, built-in help, and README.

— Names the stages but no command or config to act on v0.7.0

WAS THIS USEFUL? 👍 👎

open-codex-computer-use

Feedback ⓘ v1.1.0 NOTES ↗

RANK

Open Computer Use – Open-Source Alternative to Codex Computer Use

> Open Computer Use's two releases add macOS Locked Use for GUI automation behind a locked screen, a native MCP server with one-command installers for Codex, Claude Code, Gemini CLI, opencode and dsh, a persistent JavaScript REPL, sequenced multi-tool calls, and new diagnostics, screenshot-format and accessibility-query tooling.

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

01 Sequenced multi-tool calls via open-computer-use call NEW

95

Adds `open-computer-use call --calls '<json>'` and `--calls-file <path>` for sequenced multi-tool calls in a single process so `element_index` state is preserved across steps, defaulting to a 1s sleep between operations configurable with `--sleep`, plus `open-computer-use call <tool> --args '<json>'` to call a single MCP Computer Use tool and print the MCP-style JSON result.

Run a multi-step automation where element_index state must survive across tool calls — e.g. open TextEdit then press Return.

```
$ open-computer-use call --calls
'[{"tool":"get_app_state","args":{"app":"TextEdit"}},
{"tool":"press_key","args":{"app":"TextEdit","key":"Return"}}]'
--sleep 0.5
```

— Flags, defaults and a runnable multi-step example all given v1.0.0

02 macOS Locked Use for automation behind a locked screen NEW

90

Enables macOS Locked Use via `ocu locked-use enable`, which installs signed components, grants permissions, and runs GUI automation behind display shields while the Mac is locked — tasks relock the Mac before removing the shields.

Set up Locked Use on a macOS host so an AI agent can continue GUI automation while the Mac screen is locked.

```
$ ocu locked-use enable
```

— Names exact command and mechanism, missing limits on what tasks can run v1.1.0

03 One-command MCP installers for AI coding tools NEW

90

Adds `install-codex-mcp`, `install-claude-mcp`, `install-gemini-mcp`, `install-opencode-mcp`, and `install-dsh-mcp` subcommands that write MCP config entries directly into `~/.codex/config.toml`, `~/.claude.json`, `./.gemini/settings.json`, and `~/.config/opencode/opencode.json`, plus an `install-codex-plugin` subcommand to install open-computer-use as a Codex App plugin; `install-gemini-mcp --scope user` writes the Gemini CLI integration to the user-level config instead of the project-level `.gemini/settings.json`.

Wire open-computer-use into Gemini CLI at the user level so every project can use Computer Use without per-repo config.

```
$ npm i -g open-computer-use
open-computer-use install-gemini-mcp --scope user
```

— Every installer subcommand and target config file named v1.0.0

04 CLI diagnostics: capabilities and permission doctor checks

IMPROVED

90

Adds `ocu capabilities` and `ocu capabilities --json` to inspect Node, REPL adapter/kernel, and native runtime availability at the command line, plus `ocu doctor --json` for structured, machine-readable permission checks returning `platform`, `accessibilityTrusted`, `screenCaptureGranted`, `allGranted`, and `missingPermissions` without opening the onboarding UI (macOS).

Check all required permissions in CI or a health-check script without triggering the interactive onboarding dialog.

```
$ ocu doctor --json
```

Check whether macOS accessibility and screen-recording permissions are in place before running an agent workflow.

```
$ ocu doctor --json
```

— Exact flags and returned fields named with runnable commands v1.1.0 v1.0.0

05 Persistent JS REPL and one-shot eval with cua bindings NEW

85

Adds `ocu js '<expr>'`, `ocu js -` (stdin), and `ocu js --file <path>` for one-shot JavaScript evaluation against live `cua` bindings, closing the Worker and native child after evaluation; `ocu repl` keeps a persistent JavaScript REPL with `cua` bindings alive for the terminal session, supporting `.reset` to clear bindings and exiting on `.exit` or Ctrl-D.

— All invocation forms named but no worked example given `v1.0.0`

---

06 **Persistent screenshot format configuration** NEW 70

Adds `ocu config` for persistent screenshot settings, supporting PNG, JPG, and WebP output formats.

Persist screenshot output as WebP so all subsequent captures use that format instead of the default.

```
$ ocu config
```

— Formats and command named, no detail on default or other options `v1.1.0`

---

07 **Native MCP server with accessibility query tool** IMPROVED 65

Ships native MCP compatibility exposing nine core Computer Use tools plus a macOS-only `query` tool for finding controls by text or role without taking a screenshot; adds app.query() for targeted accessibility queries that find controls by text or role without taking a screenshot.

— Tool count and purpose named, no example of calling query `v1.1.0` `v1.0.0`

---

08 **Cross-platform npm and binary distribution** NEW 60

Distributes via `npm i -g open-computer-use` and direct download for macOS, Linux, and Windows (Linux and Windows runtimes experimental); exposes `ocu` as a short CLI alias.

— Install command given but no download links or version specifics `v1.0.0`

THINNER COVERAGE BELOW

09 **Agent-display window placement and capture on macOS** NEW 45

Adds opt-in agent-display placement on macOS with automatic restoration of window position at session end, plus covered and other-Space window capture.

— Describes behaviour but no flag or command to enable it `v1.0.0`

WAS THIS USEFUL? 👍 👎

chap

Feedback ⓘ v0.3.2 NOTES ↗ RANK

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 shipped a major storage overhaul — incremental writes, concurrent-writer conflict detection, and a restructured SQLite schema — alongside a new `create-chap-app` scaffolding tool with five project templates, profile diffing, review-desk insights, and cross-language analytics support.

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

01 **Generated CHAP project features** NEW 95

`npx create-chap-app <project>` scaffolds a full CHAP project (review desk, `POST /chap` endpoint, SQLite store, `diff-profiles` command) across five templates — code gate, MCP gate (Claude Desktop, Cursor, Claude Code), support desk, outbound approval gate (Python), and a production set with signed calls and `doctor`; generated projects add a `profiles` setting in `chap.config.json`, the `diff-profiles` command to compare outcomes under two profile sets, and a profile explorer at `docs/profile-explorer.md`; the code-gate template gates every coding-agent commit behind human review via patches, diffs, and commit-on-approval, using four trailers and a note under `refs/notes/chap`, with `verify.mjs` checking commit ranges against the team's trust policy as a pull-request workflow; the review desk adds an Insights view counting what reviewers accept, edit, and send back, broken down by model and, for code, by file; every project also gets `analytics.py`, which writes the `chap-analytics` report, evaluation cases, and a refinement page from the SQLite store, backed by `chap-analytics` 0.2.2's new `from_sqlite` and `from_json` functions; and the playground now supports `ANTHROPIC_API_KEY`, `OPENAI_API_KEY`, and `OLLAMA_URL` model providers, falling back to a scripted agent when none is set.

Bootstrap a new CHAP code-gate project that puts every coding-agent commit behind a human review step.

```
$ npx create-chap-app my-gate
cd my-gate && npm install && npm run demo
```

Set a different profile mix for a generated project by editing the single profiles key in its config file.

```json
{
  "profiles": ["core/1.0", "review/1.0", "security-signed/1.0"]
}
```

— Exact command given plus the full list of generated surfaces. Config key, command and doc path all named with an example. Mechanism and tool name given, but no exact invocation shown. Describes the view but gives no navigation or config detail. Names functions and files but lacks a runnable example. Exact env vars named, fallback behaviour stated.

v0.3.1

---

02 **SQLite schema restructured into three tables** BREAKING

85

The SQLite schema is restructured into `chap_meta`, `chap_entries`, and `chap_objects` tables, with a `chap_workspaces` view preserved for backward-compatible reads; `chap-analytics` 0.2.3's `from_sqlite` reads the new tables in a single read transaction. A database written by v0.3.1 or earlier is migrated automatically on first open (old table kept as `chap_workspaces_v1`), but v0.3.1 and earlier cannot open a migrated file (error: 'views may not be indexed') — keep a backup before upgrading if you may need to roll back.

— Table names and exact migration/rollback risk are spelled out. v0.3.2

---

03 **Store write API enhancements** NEW

77

Adds a Store.apply(change) method so stores can apply incremental writes instead of full workspace serialisation on every call, and `SqliteStore` / `MemoryStore` now raise `StoreConflictError` when a write's audit-entry count doesn't match the store's, detecting and rejecting a second coordinator writing against the same workspace.

Implement `Store.apply` on a custom store to receive only the per-call delta and avoid full-workspace writes on every call.

```javascript
class MyStore extends SqliteStore {
  apply(change) {
    // called after each coordinator call with only what changed
    super.apply(change);
    myObservability.record(change);
  }
}
```

— Names the method and shows a working implementation example. Mechanism is clear but no usage example to act on.

v0.3.2

04 on_store_error / onStoreError failure callback IMPROVED

65

Adds an `on_store_error` callback for Python, alongside the existing TypeScript `onStoreError`, so failed store writes are surfaced to the application without crashing the coordinator.

Handle store write failures in Python without crashing the coordinator — missed entries are retried on the next write.

python

```
coordinator = Coordinator(
    store=SqliteStore('workspace.db'),
    on_store_error=lambda err: logger.error('Store write failed:
%s', err)
)
```

— Callback name and a runnable code example given. v0.3.2

THINNER COVERAGE BELOW

05 MCP SDK version requirement bump BREAKING

45

`@brightbeamai/chap-coordinator-mcp` now requires `@modelcontextprotocol/sdk` 1.32.1 or later.

— Exact package and version given but minimal explanation. v0.3.1

└──▷ BREAKING ON UPGRADE

! A SQLite database file written by v0.3.1 or earlier is migrated to the new `chap_meta`, `chap_entries`, and `chap_objects` tables on first open; the old table is preserved as `chap_workspaces_v1`, but v0.3.1 and earlier will refuse to open the migrated file with the error 'views may not be indexed'. Keep a copy of any database you may need to roll back to v0.3.1.

! `@brightbeamai/chap-coordinator-mcp` now requires `@modelcontextprotocol/sdk` 1.32.1 or later.

WAS THIS USEFUL? 👍 👎

mcp

Feedback ⓘ v1.0.2 NOTES ↗ RANK

Rapidly build MCP servers for your Laravel applications.

> Laravel MCP v1.0.2 refines OAuth metadata and tightens protocol request validation, with no new client integrations in this release.

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

01 **OAuth metadata and route configuration improvements** IMPROVED 60

Laravel MCP now advertises token endpoint auth methods in OAuth metadata, and oauthRoutes() can be configured to skip dynamic client registration.

— Names the config surface but lacks usage example. v1.0.2

THINNER COVERAGE BELOW

02 **Stricter MCP request validation** IMPROVED 44

The server now rejects requests with a null `id`, and rejects legacy requests that carry an unsupported protocol version header.

— Describes behaviour change but no reproduction steps. v1.0.2

03 **Tool title included in annotations** IMPROVED 40

Tool responses now include the tool's title in `annotations.title`.

— Bare statement of a small metadata addition. v1.0.2

WAS THIS USEFUL? 👍 👎

windows-mcp

Feedback ⓘ v0.8.8 NOTES ↗ RANK

MCP Server for Computer Use in Windows

> Windows-MCP added visibility and safety controls for AI-driven sessions, including user takeover and process exclusion, plus region-scoped screen capture.

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

01 **Region capture in Screenshot and Snapshot tools** IMPROVED 55

Adds optional region capture to the Screenshot and Snapshot tools, enabling scoped screen capture of a defined area rather than the full display.

— Names exact tools and new scoped-capture behavior v0.8.6

02 **AI control visibility and user takeover** NEW 40

Adds the ability to show when AI is in control, temporarily block physical input during AI-driven sessions, and allow the user to take over at any time.

— Describes behavior but no exact UI or API surface named v0.8.8

03 **Process exclusion from UIA traversal** NEW 25

Supports excluding specific processes from UIA traversal.

— Bare capability name, no config or flag given v0.8.8

WAS THIS USEFUL? 👍 👎

task-orchestrator

Feedback ⓘ v3.16.0 NOTES ↗ RANK

MCP server enforcing workflow discipline for AI agents with persistent work items, dependency graphs, quality gates, and actor attribution. Schemas block non-compliant calls.

> task-orchestrator's biggest window yet: a new `/task-orchestrator` :run-wave skill orchestrates multi-item execution across planner, implementer and test-author seats, dispatch profiles and per-seat overrides extend trait configuration, and a new audit workflow adds adversarial review — alongside a DNS-rebinding guard, a canonical gate-status endpoint, structured JSON logging, and a round of actor-authentication hardening that now aborts startup on previously-silent misconfigurations.

└──▷ **WHAT SHIPPED · 18 FEATURES** most completely described first ? what's the number?

01 Dispatch configuration as a trait dimension NEW 90

Traits gain a `dispatch.<queue|work|review>: {agent?, model?, effort?}` profile surfaced on `advance_item`, `get_context`, query_items(operation="schema"), and `GET /config/traits`, plus a `dispatch.<phase>.seats.<seat>: {agent?, model?, effort?}` sub-key for per-seat agent, model, and effort overrides (with `agent: null` clearing a seat's phase default), a `seats:` list with per-seat `enters`, `after`, and `reads_exclude` fields served as `seats` and `dispatchBySeat` on `get_context` and query_items(operation='schema'), and new `seat` and `independent_of` fields on note-schema entries (parsed and served; enforcement deferred).

Override per-seat agent and effort for the implement phase in a trait, and clear the phase-level default for the test-author seat.

yaml

```yaml
dispatch:
  implement:
    seats:
      planner:
        agent: task-orchestrator:planner
        effort: high
      test-author:
        agent: null
```

— Config key and four surfaces named; no example shown Mechanism, served fields and config example all present

v3.15.0 v3.16.0

02 Completion gate status endpoint NEW 90

Adds the `GET /items/{id}/gate` REST route returning an item's canonical gate status (`canAdvance`, missing notes) with trait merging, per-root layering, and ETag support.

Poll whether a work-item's required notes are satisfied before attempting to advance it, using the new gate endpoint.

```
$ curl -s -H 'Authorization: Bearer <token>'
https://orchestrator.internal/items/item-42/gate
```

— Endpoint, fields and curl example all present `v3.15.0`

---

03 **Run-wave orchestration skill for multi-item execution** NEW

85

The `/task-orchestrator:run-wave` skill is a front-door orchestrator that plans a wave of ready items via `run-planner.mjs` (`probe|plan|explain|validate` subcommands), reads `.taskorchestrator/run-profile.json` for per-repo verify commands and search scope, and launches `workflows/implement-wave.js`, which runs planner, implementer, declarations-extractor, and blind test-author seats in order with concurrent multi-item execution and `dependency_blocked` / `resource_unavailable` / `config_unavailable` deferral mapping. It writes structured `delegation-metadata` provenance and ships the `task-orchestrator:planner` and `task-orchestrator:test-author` agent definitions.

— Mechanism and subcommands fully named; no invocation example shown `v3.16.0`

---

04 **DNS-rebinding protection via MCP_ALLOWED_HOSTS** BREAKING

85

Adds the `MCP_ALLOWED_HOSTS` environment variable to allowlist non-loopback `Host` headers for LAN, reverse-proxy, and Docker-network deployments; requests from unlisted hosts now receive `403`, so such deployments must set this variable to keep working.

Allow a reverse-proxy or Docker-networked host to reach the MCP/API endpoints without receiving 403 from the DNS-rebinding guard.

```
$ MCP_ALLOWED_HOSTS=orchestrator.internal,orchestrator.lan
docker compose --profile http up
```

— Env var, effect and runnable example all present `v3.15.0`

---

05 **Structured JSON logging with correlation fields** BREAKING

85

Adds the `LOG_FILE` environment variable for opt-in file logging alongside JSON-on-stderr output, with per-call MDC correlation fields (`transport`, `tool`, `sessionId`, `requestId`, `actorId` on MCP calls; `requestId`, `httpMethod`, `httpPath` on REST). Logs are now JSON on stderr by default, breaking scrapers expecting plain text, and the Docker image no longer creates `/app/logs`.

Redirect structured JSON logs to a file for ingestion by a SIEM while keeping stderr clean in production.

```
$ LOG_FILE=/var/log/task-orchestrator/app.json docker compose --profile http up
```

— Env var, fields and runnable example all present `v3.15.0`

---

06 **Actor authentication and verifier hardening** BREAKING

85

Adds `allow_insecure_url` to `actor_authentication.verifier` config to permit `http` JWKS/OIDC sources for loopback-only hosts. An unparseable global config (YAML syntax errors, unknown `degraded_mode_policy` / `verifier.type` values, a `jwks` verifier with no key source, wrong-typed verifier field) now aborts startup instead of silently falling back to schema-free / noop-verifier behaviour. A bearer token with a malformed `scope` block, or an unrecognised token-entry key (e.g. `scopes:`), now fails startup; a JWKS token with a malformed `to_scope` claim is rejected with `401 invalid_token`; a JWKS actor proof without an `exp` claim is now `REJECTED`; `jwks_uri` / `oidc_discovery` must use `https` unless `allow_insecure_url: true` is set for a loopback-only host. Under `did_allowlist` / `did_pattern` trust, a token's `sub` must equal its issuer DID. The V17 migration nulls stored raw actor proofs and replaces them with `actor_proof_sha256` / `actor_proof_claims` — rotate actor-signing keys and optionally run an offline `VACUUM` plus FTS5 rebuild before upgrading.

— Every validation change and config key named explicitly `v3.15.0`

---

07 **Config setup and discovery** NEW

75

Adds `~/.taskorchestrator/config.yaml` as the user-level config file, written by `init --user` and checked during config discovery, the `/task-orchestrator:init` skill for setting up a project root and config for the current directory or, with `--user`, a personal root, and new environment variables `TASK_ORCHESTRATOR_SETUP_HINT=off` to silence the once-per-directory init suggestion at session start, `TASK_ORCHESTRATOR_HOME` and `TASK_ORCHESTRATOR_CEILING` for config discovery isolation, and `TASK_ORCHESTRATOR_API_URL` to override the REST URL resolution used by all skills.

Set up a personal root so task-orchestrator anchors new items to your user scope without requiring a project directory.

```
$ /task-orchestrator:init --user
```

Disable the daily init suggestion in CI or shared environments where prompts are unwanted.

```
$ export TASK_ORCHESTRATOR_SETUP_HINT=off
```

— Command and resulting file path both demonstrated Four env vars named; one demonstrated with a command

v3.16.0

08 **Actor audit CLI --result flag** NEW 65

Adds a `--result` flag to the `actors` CLI subcommand of `run-exec-lib.mjs` so a Step-2 actor audit skips notes from stages that never ran.

— Flag and behavior named, no example shown v3.16.0

09 **Ralph headless-iteration and continuation handling** IMPROVED 65

Ralph iterations now set `TASK_ORCHESTRATOR_MODE=headless-iteration` so retrospective and phase-guard hooks stay silent, and new `--max-continuations` support lets Ralph resume iterations that end without a `RALPH_OUTCOME` marker instead of counting them as errors.

— Env var and flag named with concrete effect v3.15.0

10 **Item-transition error codes and selector outcomes** BREAKING 65

Every `advance_item` failure now carries an `errorCode` (e.g. `gate_blocked`, `none_eligible`) for hook and orchestrator branching. The `claim_item` selector outcome `no_match` is replaced by `queue_empty` and `none_eligible`; consumers branching on `no_match` must update their logic.

— Error codes and renamed outcomes both named explicitly v3.15.0

11 **Audit and review-wave workflow** NEW 60

Adds the `workflows/audit.js` Claude Code Workflow script with `quick|standard|full` presets, barrier-per-phase reviewers, and adversarial verification, producing a findings proposal that `/task-orchestrator:create-item` materializes after human confirmation.

— Presets and flow named; no exact command given v3.16.0

12 **Retro-analysis workflow for trend matching** NEW 60

Adds a `retro-analysis` workflow for sharded trend matching over prior retrospectives, opt-in via `/session-retrospective --deep`.

– Flag given but underlying mechanism only briefly described v3.16.0

THINNER COVERAGE BELOW

13 **Dispatch-contract rules and Method B prompt envelope** NEW 55

Adds new served rules `declarations-extractor` and `contract-change-sweep`, fetched by key in the dispatch-contract template, and a `meta.method: 'B'` plan marker that causes `run-planner.mjs prompt` to append a METHOD B RETURN envelope schema block to seat prompts, eliminating re-emit retries for Method B stages.

– Rule names and marker given; mechanism thinly described v3.16.0

14 **Phase guard plugin and registration self-check** NEW 55

Adds a SubagentStop phase guard plugin that returns a phase-owning subagent when required notes are still missing, respecting implementer/reviewer seat ownership, and a SessionStart registration self-check that warns when an MCP registration key would not match the plugin's hook matchers.

– Two plugins named; behavior described briefly v3.15.0

15 **Workflow test harness additions** NEW 45

Adds `scripts/tests/workflow-harness.mjs`, a fake-agent test harness for workflow scripts, and `scripts/tests/workflow-harness-ext.mjs` with `loadCoreNamed`, `autoAgent`, `assertBarrier`, `freeRuntimeIds`, and `ruleWindowHits` helpers for workflow scripts beyond `implement-wave`.

– Named internal helpers; no end-user usage path v3.16.0

16 **Cascade event error field** IMPROVED 40

Adds a structured `error` field to `AdvanceCascadeEvent` / `CascadeEventDto` cascade events, populated when a cascade's apply fails.

– Single field addition, minimal mechanism described v3.15.0

17 **Release gating for Docker image publishing** IMPROVED 40

Docker publishing now requires the test suite, an MCP stdio smoke test (including a stdout-purity check), and a Trivy CRITICAL/fixable scan to pass before any image is pushed.

– Pipeline steps named; no config or command shown v3.15.0

18 Lifecycle config removal and capability cleanup DEPRECATED

40

The `lifecycle: auto-reopen` config value is removed; configs using it now load as `auto` with a warning. The MCP server no longer advertises `prompts` / `resources` capabilities.

— Two removals named; minimal elaboration given v3.15.0

╰──▷ BREAKING ON UPGRADE

! Non-loopback clients now receive `403` unless their hostname is listed in `MCP_ALLOWED_HOSTS`; LAN, reverse-proxy, and Docker-network deployments must set this variable.

! An unparseable global config (YAML syntax errors, unknown `degraded_mode_policy` / `verifier.type` values, a `jwks` verifier with no key source, wrong-typed verifier field) now aborts startup instead of silently falling back to schema-free / noop-verifier behaviour.

! A bearer token with a malformed `scope` block, or an unrecognised token-entry key (e.g. `scopes:`), now fails startup; a JWKS token with a malformed `to_scope` claim is rejected with `401 invalid_token` — both previously granted unrestricted access.

! A JWKS actor proof without an `exp` claim is now `REJECTED`; `jwks_uri` / `oidc_discovery` must use `https` unless `allow_insecure_url: true` is set for a loopback-only host.

! Under `did_allowlist` / `did_pattern` trust, a token's `sub` must equal its issuer DID; tokens asserting a different `sub` are rejected.

! The V17 migration nulls stored raw actor proofs and replaces them with `actor_proof_sha256` / `actor_proof_claims`; rotate actor-signing keys and optionally run an offline `VACUUM` plus FTS5 rebuild before upgrading.

! `claim_item` selector outcome `no_match` is replaced by `queue_empty` and `none_eligible`; consumers branching on `no_match` must update their logic.

! Logs are now JSON on stderr; scrapers expecting the old plain-text format must switch to JSON parsing. The Docker image no longer creates `/app/logs`; set `LOG_FILE` for file output.

! The `lifecycle: auto-reopen` config value is removed; configs using it load as `auto` with a warning.

! The MCP server no longer advertises `prompts` / `resources` capabilities.

WAS THIS USEFUL?  

◆ AI Agent Frameworks

Arcade

Feedback ⓘ snapshot-20261025 NOTES ↗ RANK

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

> Arcade's biggest move this window is a standalone Plugin that brings its 1,000+ tool catalog into Claude, Cursor, VS Code and other agent clients without MCP gateway setup, alongside rebuilt Slack and Google Calendar toolkits (with breaking tool-name and response-shape changes), new Outlook and Linear tools, retry-safe Outlook event creation, and Kubernetes resource/egress controls for self-hosted deployments.

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

01 Kubernetes resource quota and egress controls for self-hosted deployments

NEW 85

Adds `deployments.resourceQuota.enabled` to apply a Kubernetes ResourceQuota to the namespace, and `deployments.network.egress.allow` to permit specific outbound destinations when network policies restrict egress.

Cap namespace resource consumption in a self-hosted Arcade deployment to prevent runaway workloads from exhausting cluster capacity.

yaml

```yaml
deployments:
  resourceQuota:
    enabled: true
```

Allow the Arcade worker to reach a specific external secret store when default network policies block all egress.

yaml

```yaml
deployments:
  network:
    egress:
      allow:
        - vault.internal.example.com
```

— Exact config keys with runnable YAML examples. snapshot-20261025

02 Arcade Plugin for agent clients NEW 76

The Arcade Plugin installs into Claude Code, Claude Desktop, Cursor, VS Code, GitHub Copilot CLI, and Codex, giving agents access to Arcade's 1,000+ tool catalog without configuring an MCP gateway or editing client settings. Its Tool Recommendation capability keeps the full tool catalog out of the model's context window so a single shared gateway can serve all accounts, and the Plugin ships Skills and lifecycle hooks that teach the agent when to use Arcade and surface account-authorization links the first time a service such as Gmail or Slack is needed.

Try Arcade tools from inside VS Code or Cursor the moment you sign up — no gateway config needed.

📍 Steps: 1. Install the Arcade Plugin in Claude Code, Claude Desktop, Cursor, VS Code, GitHub Copilot CLI, or Codex. 2. Sign in with your Arcade account when prompted. 3. Ask your agent something real, e.g. 'summarize my unread email' or 'post the release notes to #eng'. 4. When a service like Gmail or Slack appears for the first time, follow the in-agent link to authorize that account.

— Names every supported client and install flow, but no CLI command given.

launch-20261022-d21b34e1

---

03 **Google Calendar tools rebuilt with reminders, timezone and all-day handling**

75

BREAKING

Google Calendar's tools were rebuilt, changing tool names and several response shapes. `CreateEvent` can now set reminders directly at creation time instead of requiring a separate follow-up call, and accepts a time-without-timezone or date-only `UNTIL` value in a repeating event. List and search tools return each event's meeting link when one is attached. `UpdateEvent` can change an event's own time zone (keeping its clock time but shifting the actual instant it represents) and can switch an event between timed and all-day (and back) for a single event, one occurrence, or an entire series; both `UpdateEvent` and `RespondToEvent` now accept a broader set of inputs covering time zones, all-day/timed switching, version matching, and reply notes.

— Detailed per-tool mechanism but no exact field or endpoint names. snapshot-20261025

---

04 **Slack toolkit rebuilt with search, files, canvases and scheduling**

64

BREAKING

The Slack toolkit was rebuilt as a new major version, adding tools for message and content search, cited message/thread context, file upload and download, canvas creation and editing, typed lists, pins, reactions, and scheduled messages, with upfront OAuth-scope validation. Tool renames in the rebuild carry no alias for the old name, and

several tools return new response shapes for search, files, canvases, and lists, so code reading specific response fields may need updating.

— Lists new capabilities and breaking changes but no exact tool names. snapshot-20261025

THINNER COVERAGE BELOW

05 **Retry-safe Outlook Calendar event creation** IMPROVED

58

Outlook Calendar `CreateEvent` is now retry-safe: a repeated request returns the original event instead of creating a duplicate and re-sending invitations.

— Clear before/after behavior with named tool, no further mechanism. snapshot-20261025

06 **Outlook Mail tool for conversations awaiting reply** NEW

38

A new Outlook Mail tool lists conversations waiting on a reply without paging through every message in a mailbox.

— One behavior described, no tool name or parameters given. snapshot-20261025

07 **Reduced API calls in GitHub PR reading tool** IMPROVED

34

GitHub's pull-request reading tool now limits how many requests it makes, reducing the chance of hitting a rate limit on large pull requests.

— No request cap number or flag specified. snapshot-20261025

08 **Linear issue team move tool** NEW

24

A new Linear tool moves an issue to a different team.

— Single-line description with no further detail. snapshot-20261025

09 **Credential-free authentication to cloud secret stores** NEW

22

Enables authenticating to cloud secret stores without a long-lived credential.

— No named provider, mechanism, or config surface given. snapshot-20261025

└──▷ BREAKING ON UPGRADE

! The Slack toolkit was rebuilt as a new major version: tool renames carry no alias for the old name, and several tools return new response shapes for search, files, canvases, and lists — code reading specific response fields may need updating.

! Google Calendar's tool names and several response shapes changed as part of its rebuild.

WAS THIS USEFUL? 👍 👎

magic-context

Feedback ⓘ v0.49.21 NOTES ↗ RANK

Unbounded context. Memory that manages itself.

> Magic Context's v0.48–v0.49 line adds Synapse-backed reranking to `ctx_search`, a `smart_notes.enabled` kill switch, and experimental OpenCode 2 transcript compaction, while the dashboard ships a light theme, a redesigned Cache page, a structured Protected Tools table, and support for new Broca log record types, alongside deprecations of `models.window_overlay_path` and `mural.model`.

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

01 **Deprecated config keys** DEPRECATED 95

`models.window_overlay_path` is deprecated and ignored — configs that still set it load with a deprecation warning and the key should be removed — and `mural.model` (including `experimental.mural.model`) is deprecated and ignored in favor of `dreamer.<harness>.tasks.compress-cues.model` where `<harness>` is `opencode`, `pi`, or `omp`; in the dashboard the cue compressor is now picked from each host's dreamer settings instead of a separate shared field, though `mural.model` is still honoured as a legacy key editable in Raw JSONC for Magic Context 0.47.x.

— Exact config key named with clear removal instruction Exact deprecated and replacement keys with migration path

v0.49.1 dashboard-v0.20.0 v0.48.0

02 **smart_notes.enabled config switch** NEW 95

Adding `smart_notes.enabled: false` to user config disables condition-based note authoring and evaluation while keeping ordinary and already-ready notes available. Changing this setting requires a restart and must be set only in user config, not project config.

Disable smart/condition-based note authoring for a session while retaining ordinary notes.

```yaml
smart_notes:
  enabled: false
```

— Exact config key plus runnable example and restart caveat v0.48.0

03 **Synapse reranking for ctx_search** NEW 90

`ctx_search` reorders natural-language query results using the Synapse ranking service when available, with automatic fallback to previous ranking for symbol/path lookups, search failures, or installs without Synapse. Hosted reranking can be configured via a

`rerank` block in `~/.config/cortexkit/magic-context.jsonc` (or the Windows/XDG equivalent); local reranking remains the default.

— Names config key and file path but no runnable command v0.49.0 v0.49.21

04 **Experimental transcript compaction via MC_OC2_INVISIBLE_FOLD**

NEW 85

Setting the `MC_OC2_INVISIBLE_FOLD=1` environment variable enables experimental transcript compaction on OpenCode 2, reducing long-session delays without changing model requests.

Enable experimental transcript compaction on OpenCode 2 to cut delays in long sessions.

```
$ MC_OC2_INVISIBLE_FOLD=1 opencode
```

— Exact env var and runnable command provided v0.48.0

05 **magic-context doctor --verbose flag** NEW 85

The `magic-context doctor --verbose` flag prints the full dangling-compartment listing, whereas ordinary output shows only counts and up to ten examples.

Inspect every dangling compartment (not just the first ten) when diagnosing stale state.

```
$ magic-context doctor --verbose
```

— Exact flag with runnable command v0.48.0

06 **Dashboard UI improvements** IMPROVED 80

The dashboard replaces the `protected_tools` raw JSON box with a structured table showing tools, how many newest results to keep (built-in defaults of newest todo list and newest three `ctx_reduce` calls), and inline validation; historian rows now display the provider and fallback attempts used for a run, with errors shown as a red icon with hover details; Pi session pages now show compartment dates, with aligned session headers and background-model tabs; the Config page adds a Keep reasoning tokens setting, replacing the old reasoning-age setting, matching Magic Context 0.47.0; the Cache page is redesigned so session cards show readable names instead of raw IDs, the timeline is drawn to scale against the model's context window with a fill bar, and turns are rendered as a table, while Live mode now reads only changed data since the last refresh instead of rescanning the full database and logs on every poll, dropping a Broca-filtered refresh from ~627 ms / 454 MB read to under 5 ms; and the Logs view pauses refreshing while the window is hidden, slows down when unfocused, and re-reads only log files that changed.

— UI path and field name given, no exact command UI path given, limited mechanism detail Thin UI description, no mechanism beyond display UI path named, no further mechanism given Concrete before/after numbers but only a UI path to use it Automatic performance change, nothing to configure

dashboard-v0.20.0 v0.48.0

07 **Dashboard: new log record support, memory restore, and light theme**

NEW 75

Dashboard 0.20.18 adds support for three new Broca log record types — `provider-declarations/v1` (plan manifest with declared result formats and `late_results` capability), `provider-held/v1` (held tool-call records and a `tool_held` pause reason), and `plan-reference-slots/v1` (reference slots from a run's plan manifest) — so run steps after them are no longer silently dropped; adds a Restore button that reinstates dismissed user memories into the active set, with running sessions picking up the restored entry immediately; and adds a light theme with a System/Light/Dark picker in the sidebar, with System following the OS as the default.

— Internal format fix, nothing for a user to invoke UI path given but no config or endpoint UI path given, straightforward toggle

dashboard-v0.20.18 v0.49.1 dashboard-v0.20.0

THINNER COVERAGE BELOW

08 **ctx_reduce scoped to requested tags** IMPROVED 55

`ctx_reduce` now checks only the requested tags instead of scanning the full session, reducing per-call overhead in large sessions.

— Clear before/after but no user-facing control v0.49.0

09 **ONNX Runtime telemetry opt-out fixed** IMPROVED 50

The ONNX Runtime telemetry opt-out is now effective for local embeddings under Bun and Node, including compiled Bun hosts; on Windows a one-time log line explains that OS-level event tracing is used instead.

— Automatic behavior fix, nothing for reader to do v0.49.1

10 **Longer historian/dreamer transcript retention** IMPROVED 45

Historian and dreamer transcripts in OpenCode 1 and 2 Rust mode now request up to one day's retention where the model runner supports it.

— Automatic change, no lever for the reader v0.49.0

11 **Per-turn machine-checkout check removed** IMPROVED 45

Magic Context no longer refuses turns because another machine claims the checkout; the per-turn machine-checkout check (MC-C16) has been retired.

— Behavior change explained but nothing to configure `v0.48.0`

╰──▷ BREAKING ON UPGRADE

! `models.window_overlay_path` is deprecated and ignored; configs that still set it load with a deprecation warning — remove the key.

! `mural.model` (including `experimental.mural.model`) is deprecated and ignored; use `dreamer.<harness>.tasks.compress-cues.model` where `<harness>` is `opencode`, `pi`, or `omp`.

! Changing `smart_notes.enabled` requires a restart and must be set only in user config, not project config.

! The cue compressor model is no longer a separate shared field; it is now sourced from each host's dreamer settings. An existing `mural.model` key still applies and can be edited in Raw JSONC, but the standalone shared field is gone.

WAS THIS USEFUL? 👍 👎

platypus

Feedback ⓘ v3.28.0 NOTES ↗ RANK

Self-hosted AI Agents for your whole team — on your infrastructure, your models, around the clock.

> Platypus broadened model-provider flexibility (Ollama, vLLM, OpenAI-compatible endpoints), introduced a Tool gate that requires user approval before tool calls execute, and reworked Skills into zip-based bundles with a `readSkillFile` function, alongside wide-ranging Workspace list, Trigger, and Chat UI improvements across 12 releases.

└──▷ **WHAT SHIPPED · 23 FEATURES** most completely described first ? what's the number?

01 Skill bundles as zip files with readSkillFile IMPROVED

80

Stores Skills as file bundles with zip import, export, and a `readSkillFile` function, replacing single-file skill storage; bundled Skills are copied into the Sandbox on load so they are immediately available in the execution environment.

— Names readSkillFile function and bundle format with before/after v3.19.0

02 fsView tool in Sandbox Tool set NEW

75

Adds `fsView` to the Sandbox Tool set, enabling the model to inspect file contents during sandbox execution.

When running a sandbox agent task that needs to verify or read a file's contents without executing it — use fsView from the Sandbox Tool set.

> 📍 In the Agent or Sandbox configuration, add `fsView` to the enabled Sandbox Tool set so the model can read and inspect files during its run.

— Named tool with exact steps to enable it v3.21.0

03 **Tool gate approval prompts and extension point** NEW

70

Adds a Tool gate that prompts the user to approve a tool call before it executes, plus a backend Tool gate Extension point that controls access to tools at the extension layer.

Require explicit user approval before any tool call executes, preventing automated agents from taking unreviewed actions.

📍 In the agent configuration UI, navigate to the Tool gate settings and enable the approval prompt so each tool call is held until a user confirms it.

– UI steps given but no exact config key or flag v3.28.0 v3.27.0

04 **Cloning for Shared Agents and Triggers** NEW

65

Enables cloning a Shared Agent into a Workspace from the frontend, and adds the ability to clone a Trigger directly from its card menu in the UI.

Clone a community-shared agent into your own Workspace to customise it without affecting the original.

📍 In the platypus frontend, open a Shared Agent, then select `Clone into Workspace` to create an editable copy under your account.

Duplicate an existing Trigger configuration without rebuilding it from scratch.

📍 In the Workspace home, open the Triggers list, open the card menu for the Trigger you want to copy, and select `Clone`.

– Exact UI navigation given for both clone actions v3.28.0 v3.25.0

05 **Workspace home list pagination, filters, and Trigger status**

IMPROVED

65

Adds pagination to the Workspace home Agents, Skills, Boards, and Triggers lists; keeps Skills and Trigger list filters in the URL so filter state is shareable and survives navigation; adds search and sort to the Agent form's Skills card with clamped descriptions; adds name and agent filters to the Workspace home Skills list; adds type-based filtering to the Workspace home Triggers list; adds search by name to the Workspace home Boards and Dashboards lists; shows each Trigger's last run status in the Workspace home Triggers list; enables opening a Trigger run directly from its row with actions moved into a menu; and moves the Trigger run status badge under the Trigger name.

– Enumerates six list UX additions without exact UI paths Names specific UI relocations but no exact navigation path

v3.25.0 v3.24.0 v3.23.0 v3.21.0 v3.20.0 v3.18.0 v3.19.0

06 **Ollama, vLLM, and OpenAI-compatible endpoint support** NEW

60

Platypus now supports Ollama, vLLM, and any other OpenAI-compatible endpoint through a base-URL override, in addition to connecting to OpenRouter directly.

– Names providers and override mechanism but no exact config key `product docs`

07 **AWS default credential chain for S3 storage** IMPROVED

60

Supports the AWS default credential chain for S3 and builds storage at boot, removing the need for explicit credential configuration.

– Explains mechanism but no exact env var or flag named `v3.28.0`

08 **GitHub-based Skill install, updates, and org gating** NEW

60

Checks GitHub-installed Skills for available updates, supports installing Skills from GitHub behind an Org import gate, and links each Skill to its GitHub directory in the UI during install.

– Names the GitHub install and gating flow but no exact UI path or endpoint `v3.28.0` `v3.27.0`

THINNER COVERAGE BELOW

09 **Workspace settings provider and model display** IMPROVED

55

Names the Provider on the model picker when a label is shared across multiple providers, shows provider models as cards edited in an Add/Edit model dialog, adds each provider's model ID to Workspace settings provider selects, displays a model row's number limits side by side, and adds a ⋮ menu, Disabled badge, and second line to Workspace settings lists.

– Lists five UI tweaks but no exact navigation or config `v3.28.0` `v3.23.0` `v3.21.0`

10 **Chat rendering and interrupt enhancements** NEW 55

Adds Escape key support to stop an in-flight Chat turn from the frontend UI, groups adjacent tool calls into a collapsible Tool call group in the conversation view, renders Mermaid diagrams inline in Chat, and adds syntax highlighting for code blocks in Chat.

– Describes four distinct UI behaviors, actionable via normal chat use `v3.26.0` `v3.23.0` `v3.20.0`

11 **A2A endpoint and task reliability signals** NEW 55

Flags A2A endpoints that have no live tokens, surfacing credential gaps in the UI; notifies A2A clients when a completed Task's reply was cut short so callers can detect

truncated responses; and marks A2A Chats in the sidebar with a distinct icon.

— Names three A2A-specific signals without API detail v3.20.0 v3.18.0

12 **Pinning Agents in a Workspace** NEW 55

Enables pinning Agents in a Workspace, with Pinned agents displayed in a dedicated group separate from the rest.

— Explains behavior with a clear UI starting point v3.20.0

13 **One-off Triggers created by Agents** NEW 50

Enables Agents to create One-off Triggers that hide once fired and expire automatically.

— Describes mechanism (hide, expire) without setup detail v3.18.0

14 **Background LLM call and file inlining caps** IMPROVED 45

Caps background LLM calls and tool result size to bound resource usage, and streams stored files while capping what a single turn inlines to limit per-turn memory pressure.

— Describes mechanism but no concrete limits or config named v3.28.0

15 **Frontend error boundaries and page titling** IMPROVED 45

Notifies users in the UI when a Chat session opens on a substitute model instead of the requested one, adds error boundaries for the root, segments, and chat message parts to isolate and surface frontend failures gracefully, titles Chat, Agent, Trigger, and Organization pages dynamically by their name, and titles every page while excluding the app from search indexes.

— Explains four UI robustness changes without configuration surface v3.22.0

16 **Exa web-search backend plugin** NEW 40

Adds Exa Web-search backend plugin, bringing live web-search capability to the backend.

— Names the plugin but no setup or config details v3.26.0

17 **Dashboards deprecated ahead of v4.0.0** DEPRECATED 40

Deprecates Dashboards ahead of their removal in v4.0.0.

— Names the exact removal version but no migration guidance v3.24.0

18 **Workspace transfer to another member** NEW 35

Supports transferring a Workspace to another member.

— Names the capability without mechanism or UI path v3.18.0

19 **Sub-Agent token usage counted in parent stats** IMPROVED

30

Counts Sub-Agent token usage in the parent run's stats for unified usage tracking.

— Bare description of a stats rollup, no detail on where to view it v3.28.0

---

20 **Notification source tracking for Chat and Trigger runs** IMPROVED

30

Records a Notification's source Chat or Trigger run for traceability.

— Bare description of traceability metadata v3.18.0

---

21 **Warning for unattached Organization resources** NEW

30

Warns when an Organization resource is attached to no workspaces.

— Names the condition but no remediation steps v3.23.0

---

22 **Post sign-in redirect to originally requested page** IMPROVED

25

Returns users to the originally requested page after sign-in.

— Minimal description of a navigation fix v3.23.0

---

23 **Unsaved changes warning in editors** NEW

25

Warns before discarding unsaved changes in any editor.

— Generic description without scope of which editors v3.23.0

WAS THIS USEFUL? 👍 👎

Graphicle AI Graphicle

Feedback ⓘ v0.1.7 NOTES ↗

RANK

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

> Across v0.1.5 and v0.1.7, Graphicle adds a passage index and saved code graph that cut question latency 9x, new YAML and multigraph code-graph parsing, `GRAPHICLE_BACKEND` provider selection, large-graph (>512MiB) handling, deeper language-specific parsing across ten languages, Git post-rewrite hook installation hardened against symlink attacks, and a run of smaller CLI, SDK, doctor and prompt-hook improvements.

└──▷ **WHAT SHIPPED · 18 FEATURES** most completely described first ? what's the number?

01 **Support for graph.json over 512 MiB** IMPROVED 93

Enables `graphicle graph update` on projects whose `graph.json` exceeds 512 MiB (e.g. kubernetes); the engine now self-reads its own output files over the cap for `update`, `watch`, `serve`, `query`, `explain`, `path`, `merge-graphs`, `cluster-only`, and `label`. `GRAPHICLE_MAX_GRAPH_BYTES` applies globally when set. A graph over 512 MiB written by an earlier release is not recognized as the engine's own until rebuilt once with `GRAPHICLE_MAX_GRAPH_BYTES=<N>GB graphicle graph extract --force <project>` (add `--code-only` when no LLM backend is configured).

— Exact command, env var and migration path given v0.1.7

02 **Passage index and saved code graph caching** NEW 91

Adds a passage index per project (stored under `gf-cache/<project>/query/`) used by `graphicle context`, the memory server's `recall_project`, and the prompt hook, replacing per-question full-file reads with incremental syncs that read only changed files. The code graph and its search index are saved to the same `gf-cache/<project>/query/` location after each build so questions read the cache instead of rebuilding from the store, cutting question latency 9x on large projects. `GRAPHICLE_PASSAGE_INDEX=0` forces the full-path read and `GRAPHICLE_SAVED_GRAPH=0` forces reading the code graph from the store instead of the saved index.

— Names cache paths, env vars and the 9x benchmark v0.1.5

03 **graphicle doctor and setup Questions diagnostics** IMPROVED 78

`graphicle doctor` adds a Questions check reporting whether the passage index is built and current and whether the code graph is saved, with `graphicle setup`

surfacing the same status in its `questions` field. Response packets now include a `passage_index` field indicating what the responder read, and projects exceeding the passage-index bounds (50,000 documents or 40,000 passages) are now ranked in full instead of only on the best-matching documents.

— Names exact fields and numeric index bounds v0.1.5

04 **GRAPHICLE_BACKEND LLM provider selection** NEW 75

Adds the `GRAPHICLE_BACKEND` environment variable to choose the LLM provider, with `none` disabling LLM work entirely; the auto-detected provider is named before any data is sent.

— Exact env var, values and behavior given v0.1.7

05 **Language-specific code graph parsing improvements** IMPROVED 75

C# gains method overload support (each overload its own node with ids salted by signature, calls bind by argument count, and `delegate` is now a type); adds type alias and declaration nodes for Python `TypeAlias`, `type X =`, and `NewType`, Go `type A = B`, Rust `type` aliases and `union` items, C++ namespace-scope `using A = B;` and `typedef`, Kotlin `typealias`, Dart `typedef`, and Swift `macro` declarations and #macro(...) calls; SQL parsing now unbrackets T-SQL names (e.g. `dbo.Alpha`), records `reads_from` and `writes_to` edges for procedures, treats `GO` as a statement terminator, and folds forward-referenced tables onto their `CREATE TABLE`; and Solidity gains inherited call, `super` call, and modifier resolution through C3 linearization.

— Enumerates every language and construct named v0.1.7

06 **Git post-rewrite hook install** NEW 75

Adds Git post-rewrite hook support via `graphicle-graph hook install` so a rebase, `pull --rebase`, cherry-pick, or revert triggers a single graph rebuild.

— Exact command and triggering git operations given v0.1.7

07 **Prompt hook time budget config** NEW 73

`graphicle config set prompt_context.seconds N` (1–25 s, default 10) and the `GRAPHICLE_PROMPT_CONTEXT_SECONDS` env var configure the prompt hook's time budget.

— Exact command, env var and value range given v0.1.5

08 **Extension context re-packing on token overflow** IMPROVED 73

Extends the extension interface so a `context` result exceeding an installed extension's token limit is re-packed with a smaller budget (up to 4 rounds, down to 256 tokens, reported in `response_sized`) instead of being refused with HTTP 413.

– Mechanism and before/after given, but no direct user action v0.1.5

09 **YAML structural parsing in code graph** NEW 73

Adds YAML structural parsing to the code graph: `.yaml` and `.yml` files are read as code with one node per key, `contains` edges, nesting capped at seven levels and 500 keys per file; YAML held as data is skipped by design and unsafe tags are refused.

– Concrete limits and rules, but no CLI invocation shown v0.1.7

10 **Symlink attack hardening for graph install and hooks** IMPROVED 73

Hardens `graphicle graph install` and `graphicle graph hook install` against symbolic-link attacks: skill files, `references/` staging (now a randomly named private folder), and `SKILL.md.bak` / `.graphicle-bak` backups never follow symlinks, while `update --no-cluster` and the GraphML export replace a link at their output rather than writing through it.

– Names each hardened surface and its exact protection v0.1.7

11 **Python SDK retry_after on GraphicleError** NEW 70

Adds `GraphicleError.retry_after` to the Python SDK, exposing the service's Retry-After delay in seconds (parsed from a numeric seconds value or an HTTP date), so callers can detect budget exhaustion and know when to retry.

– Named field and parsing behavior, no code sample v0.1.7

12 **Upgrade triggers one-time re-extraction and graph rebuild** 70

BREAKING

Upgrading changes the AST cache format, so the first sync after upgrading re-extracts every file once; a code graph episode is now named by the engine's graph format as well as the project's files, so the first sync also rebuilds the graph even when no file changed; and the updated semantic extraction prompt causes cached semantic results to be extracted again once, incurring one round of LLM calls.

– Clear migration impact described, no workaround given v0.1.7

13 **Multigraph support in graph.json** NEW 65

Adds `graph.json` support for parallel edges between two nodes (`multigraph: true`), with backward-compatible reading of files written under the old `multigraph: false` format across all readers including `export` and `merge-driver` .

– Names the config key and affected readers v0.1.7

14 **CLI query diagnostics and error reporting** IMPROVED 65

Adds `--help` listing of every `--platform` value; `query` now exits with code 3 when a name is shared by multiple nodes and code 4 for an unknown `--seed`; and `explain`, `get_node`, and `get_neighbors` now report a substring near-miss instead of silently answering with it.

— Exact exit codes and affected commands named v0.1.7

---

15 **Expanded loopback HTTP access** IMPROVED 60

Expands plain `http://` loopback access in the `graphicle` program to the full `127.0.0.0/8` range (e.g. `127.0.0.2`), IPv4-mapped loopback (e.g. `[::ffff:127.0.0.1]`), and `[::1]`, for local models, connectors, OAuth, vector and graph endpoints.

— Concrete address ranges named, no usage walkthrough v0.1.7

THINNER COVERAGE BELOW

16 **Plain-language location query recognition** IMPROVED 50

Plain-language location queries such as "Which file and functions compute this?" or "Where is that latency recorded?" are now recognised as code questions by the prompt hook, which answers from the code graph for the names mentioned in the prompt rather than reading all project files; this applies even while the passage index is not yet ready for a project of 2,000+ files.

— Describes behavior with examples but no config surface v0.1.5

---

17 **Windows executable version metadata** IMPROVED 35

Windows executables `graphicle.exe` and `graphicle-graph.exe` now carry version metadata (product name, company, description, version) visible in file properties.

— Small, self-contained, no further mechanism v0.1.5

---

18 **Network contact documentation** NEW 35

Adds `docs/code-signing.md` listing every case in which Graphicle contacts the network, what each contact sends, and how to disable or avoid it.

— Names the doc but not its contents v0.1.5

└──▷ BREAKING ON UPGRADE

! The AST cache format changed; the first sync after upgrading re-extracts every file once.

! A code graph episode is now named by the engine's graph format as well as the project's files, so the first sync after upgrading rebuilds the graph even when no file changed.

! A `graph.json` over 512 MiB written by an earlier release has no record of being the engine's own; `update` and hook rebuilds refuse it until it is rebuilt once with `GRAPHICLE_MAX_GRAPH_BYTES=<N>GB graphicle graph extract --force <project>` (add `--code-only` when no LLM backend is configured).

! Cached semantic results are extracted again once after upgrade due to the updated semantic extraction prompt, incurring one round of LLM calls.

! Graphicle 0.1.4 and earlier refuse a configuration containing `prompt_context`; after running `graphicle config set prompt_context.seconds N`, do not run an older Graphicle (such as an MCP server started before the upgrade) against the same data folder.

WAS THIS USEFUL? 👍 👎

atomic-agent

Feedback ⓘ v0.6.11 NOTES ↗ RANK

Atomic Agent is a local-first AI agent. Runs open-weight models on your own machine via llama.cpp.

> Atomic Agent moved local model execution onto a new Atomic Chat Core engine, unified terminal and desktop state under a single `~/.atomic-agent` directory, and added multilingual semantic memory search via EmbeddingGemma 2 alongside a new Studio for local and cloud image/video generation, while hardening shell and MCP command execution against API key leakage.

└──▷ **WHAT SHIPPED · 26 FEATURES** most completely described first ? what's the number?

01 Multilingual semantic memory via EmbeddingGemma 2 NEW

98

Adds `atomic-agent models pull-embedding embeddinggemma-2`, `models use-embedding embeddinggemma-2`, and `models start-embedding` commands to enable EmbeddingGemma 2 (310 MB, 100+ languages) for semantic/hybrid memory search, so a note in one language can be found by a query in another. `atomic-agent memory reindex` re-embeds notes written before embeddings were enabled or before switching embedding models. `memory.embeddings.fts5Weight`, `vectorWeight`, and `bruteForceCeiling` config keys in `config.json` tune hybrid search ranking. In the desktop app, selecting `EmbeddingGemma 2` under Settings › Models › Embedding models while a local chat model is active enables the same cross-language semantic search.

Enable multilingual semantic memory search so notes and queries in different languages match by meaning.

```
$ atomic-agent models pull-embedding embeddinggemma-2
atomic-agent models use-embedding embeddinggemma-2
atomic-agent models start-embedding
```

After switching to EmbeddingGemma 2, force all existing notes to be re-embedded immediately rather than waiting for background indexing.

```
$ atomic-agent memory reindex
```

Tune hybrid search to weight vector similarity more heavily than full-text keyword matching.

```json
{
  "memory": {
    "embeddings": {
      "fts5Weight": 0.2,
      "vectorWeight": 0.8,
      "bruteForceCeiling": 500
    }
  }
}
```

Enable meaning-based memory so the agent retrieves notes by semantic similarity and across languages during a local-model session.

📍 Settings › Models › Embedding models — select EmbeddingGemma 2, then start or continue a chat with a local model active.

— Exact commands, config keys, and model size given with runnable examples. v0.6.8 desktop-v0.0.2

02 **API key isolation from shell and MCP commands** BREAKING

91

Adds `tools.shell.envPassthrough` config key to explicitly allowlist environment variables that shell commands may inherit — all other variables matching key/token/secret/password naming patterns are stripped before execution. MCP servers started from a command now receive only basic system variables plus their own config `env`, not all provider keys. In the desktop app, shell commands no longer receive API keys either, and MCP servers receive only the keys in their own config.

Prevent API keys from leaking into shell commands while still allowing a specific token a script genuinely needs.

```yaml
# In your atomic-agent config
tools:
  shell:
    envPassthrough:
      - GITHUB_TOKEN
      - NPM_TOKEN
```

— Names exact config key and stripping rule, with a runnable example. v0.6.11 desktop-v0.0.5

03 **Unified state directory across terminal and desktop apps**

91

BREAKING

Unifies state under `~/.atomic-agent` so the terminal agent and desktop app share chats, memory, skills, keys, and settings, and can run simultaneously without one stopping a local model used by the other. Adds `atomic-agent import atomic-desktop --source <dir>` to migrate chats, scheduled tasks, and memory from an older desktop folder. On first launch after upgrade, desktop app data from `~/.atomic-agent-desktop` is migrated to `~/.atomic-agent`; where both apps have the same setting, the terminal agent's value wins and overwrites the desktop value. `atomic-agent uninstall` now warns when it detects that the desktop app shares the same data folder.

Migrate an older Atomic Desktop data folder into the unified `~/.atomic-agent` state directory before running both apps side-by-side.

```
$ atomic-agent import atomic-desktop --source <dir>
```

— Exact paths, command, and migration precedence documented. v0.6.11 v0.6.9 desktop-v0.0.3

04 **Studio for local and cloud image/video generation**

NEW INTEGRATION NEW

88

Adds a Studio sidebar group with Images and Videos pages (stored in `models/studio/images` and `models/studio/videos`), with image-generation modes Create, Edit, Reference, Transform, Inpaint, Extend, and Upscale (each gated per model support) and controls for shape, quality, count, plus an Advanced fold for negative prompt, steps, guidance, and seed. Video generation runs from text or a first frame at 4, 6, or 8 seconds (default 4) in 16:9 or 9:16, with sound, an inline player, and a clip library. Supports cloud image/video models via OpenRouter (showing price per image or per second when published) and local Studio models from the Atomic Chat catalog with checksum verification and free-disk-space checks, plus labels for non-commercial and 18+ models. The agent can also generate images/video directly from chat with inline playback and an Open in Studio action.

— Extensive mechanism and modes named; no CLI/API surface given. desktop-v0.0.4

05 **Atomic Chat Core local model engine**

BREAKING

85

Switches local model execution to Atomic Chat Core, which downloads and manages `llama.cpp` automatically and fits context size and GPU placement to the host machine; existing setups migrate automatically on first start (config v77), and `atomic-agent models engine llama-server` reverts to the legacy engine. The desktop app ships Atomic Chat Core 0.18.6 in the installer and bundles it as its model engine.

– Names the engine, version, migration, and revert command. desktop-v0.0.4 v0.6.8 desktop-v0.0.2

06 **Command palette with fuzzy search** NEW 84

Adds a `Cmd+K` command palette with fuzzy/typo-tolerant search (e.g. `mdl` matches Models) that surfaces up to 20 chats ranked by best match, plus `Cmd+=` / `Cmd+-` zoom in/out and `Cmd+0` Actual Size; the sidebar toggle moved to `Cmd+\` (or `Ctrl+\` on Windows/Linux).

– Exact shortcuts and behaviour named for every control. desktop-v0.0.5 desktop-v0.0.4

07 **MCP server configuration and tool filtering** NEW 78

Adds `enabledTools` and `disabledTools` fields to MCP server config to expose only a subset of an MCP server's tools to the agent. The desktop app adds MCP smart-add: paste a server's command line, URL, or raw JSON to register an MCP server.

Limit an MCP server to only the tools relevant to your workflow, reducing the agent's attack surface.

yaml

```yaml
# In your atomic-agent config, under the relevant MCP server
entry
mcpServers:
  my-server:
    command: npx my-mcp-server
    enabledTools:
      - read_file
      - list_directory
```

– Names exact config fields and the desktop registration method. v0.6.11 desktop-v0.0.5

08 **Telegram pairing via one-time code** NEW 75

Telegram pairing now requires a 6-digit one-time code (valid 5 minutes) sent via `/start <code>`, and a paired owner is only replaced after explicit confirmation. The desktop app generates the same one-time code directly in-app and accepts `/start <code>` within the 5-minute window, with per-setting Restart agent buttons.

– Full mechanism and timing given, no runnable snippet. v0.6.11 desktop-v0.0.5

09 **Command approval flow improvements** NEW 75

Adds `allow-in-chat` to the local API so a client can approve a program (e.g. `git`) for the remainder of one chat session without persisting the grant. Approval cards now expand multi-line commands, showing (+N more lines) and opening a full preview.

— Names the API addition and the UI behaviour change. v0.6.11 desktop-v0.0.5

10 **File, image, and archive attachments in chat** NEW 75

Adds file and image attachment support via the + button, drag-and-drop onto the window, or paste, with attachments shown as chips above the message box; agent-returned images render inline with a full-screen view, Save As, and Show in Finder (Show in Explorer on Windows). Also adds `zip`, `tar`, `gz`, and `tgz` archive attachment support in chat.

— Names formats and UI mechanisms for attaching and viewing. desktop-v0.0.4 desktop-v0.0.3

11 **Desktop local model management UI** NEW 75

Adds a memory model card showing live state (Starting, Running, Not running, Off, or Not downloaded), a Too big for this Mac guard that prompts before downloading or starting a model that exceeds available memory and suggests a model that fits, a `llama.cpp` custom server port suggestion of `9931` (the default of current `llama-server`), and manual GGUF import by file path: press `a` in the Local models panel and paste the path — no copy or download occurs.

— Names each UI element and the default port value. Exact keystroke and panel location given.

desktop-v0.0.5 v0.6.8

12 **New slash commands for prompt rewriting and context compaction** NEW 70

Adds `/enhance` TUI command to rewrite a draft prompt and place it back in the composer for editing before sending, and adds `/compact` command plus a Compact action in the context popover, both showing progress and token counts before and after compaction.

Reduce context size mid-session and see exactly how many tokens are reclaimed before committing.

```
$ /compact
```

— Exact command given but little further mechanism. Exact command given with a usage example.

v0.6.11 desktop-v0.0.2

13 **Expanded project rules file support** IMPROVED 66

The agent now reads project rules from `GEMINI.md`, `.cursorrules`, `.github/copilot-instructions.md`, and `.cursor/rules` in addition to existing sources.

— Names every supported file path, no mechanism beyond reading. v0.6.11

14 **Windows on ARM (Snapdragon) desktop installer** NEW 65

Adds a Windows on ARM (Snapdragon) desktop installer (`Atomic-Agent-Setup-0.0.3-arm64.exe`), running local models on CPU.

— Exact installer filename and platform named. desktop-v0.0.3

15 **`os.fs.move` built-in file tool** NEW 64

Adds `os.fs.move` built-in tool to move and rename files without invoking a shell.

— Exact tool name given, minimal further detail. v0.6.9

16 **New integrations and cloud provider presets** NEW INTEGRATION NEW 63

Adds an Exa key integration for web search (inline in chat when the free quota is exhausted, and also available in Settings › General), a Composio integration configurable from Settings › Connections in the desktop app, and new cloud provider presets — Claude Code and Codex subscription options under Add provider plus DashScope (Qwen) under Settings › Models — with cloud providers now displaying their full live model list and a filter field.

— Names the exact settings location, no further mechanism. Names exact settings location, no further mechanism. Names each new preset and the browsing location.

desktop-v0.0.4 desktop-v0.0.3 desktop-v0.0.5 desktop-v0.0.2

17 **Chat list management actions** NEW 62

Chats can be renamed via the chat menu or by double-click. A new '...' per-chat context menu (also via right-click) adds Pin, Mark as read or unread, and Delete actions; pinned chats appear in their own sidebar section.

— Names exact menu actions and their UI location. desktop-v0.0.4 desktop-v0.0.2

THINNER COVERAGE BELOW

18 **Failed turn error details and retry** IMPROVED 58

Failed turns now display a short reason, the raw error under a Details panel, and a Try Again button; provider HTTP statuses are translated to plain text on the waiting strip.

— Describes the UI mechanism fully, names no API. desktop-v0.0.3

19 **Project-specific AGENTS.md instructions in cloud workspaces** NEW 54

Cloud workspaces now load a project's own `AGENTS.md` instructions and skills, with a size-limited folder-root fallback outside git projects.

— Names the file and fallback behaviour, no exact size limit given. v0.6.8

20 **Piped input queuing during active turns** IMPROVED 53

`atomic-agent run` now queues piped lines that arrive during an active turn instead of silently dropping them.

— Names the command but offers little further detail. v0.6.11

21 **Fusion planner direct web access** IMPROVED 50

Enables the Fusion planner to call `os.web.fetch` and `os.web.search` directly.

— Names the tools called but no further mechanism. v0.6.9

22 **Faster background context compaction** IMPROVED 50

Context compaction now maintains a running background summary so compaction completes in seconds rather than triggering a chain of retries.

— Explains mechanism and before/after but names no command. v0.6.9

23 **Full tool output saved to file when clipped** IMPROVED 44

When a long tool result is clipped, the full output is now saved to a file so the model can retrieve only the portion it needs.

— Explains behaviour but names no file path or command. v0.6.11

24 **Code block and reasoning display in chat** NEW 43

Code blocks now show headers with the detected language and a Copy button; Reasoning rows display how long the model spent thinking.

— Describes two thin UI additions without further detail. desktop-v0.0.4

25 **Embedding-only provider without chat model** IMPROVED 40

An embedding-only OpenAI-compatible provider no longer requires a configured chat model.

— Describes the fix but names no config or endpoint. v0.6.9

26 **Agent reply files served over HTTP** IMPROVED 33

Files attached to an agent reply are now served to HTTP clients, not only Telegram and Discord.

— States the change but names no endpoint or config. v0.6.9

└──▷ **BREAKING ON UPGRADE**

! Shell commands no longer inherit environment variables whose names match key/token/secret/password patterns — commands that previously relied on those variables

must have them explicitly listed in `tools.shell.envPassthrough` or they will not receive them.

! On first launch after upgrade, desktop app data from `~/.atomic-agent-desktop` is migrated to `~/.atomic-agent`; where both apps have the same setting, the terminal agent's value wins and overwrites the desktop value.

! Local models now run through Atomic Chat Core instead of the previously selectable engine; existing setups are migrated automatically on first start (config v77). Use `atomic-agent models engine llama-server` to revert.

v0.6.10 NOTES ↗ STABLE

Adds image/video generation tools, archive extraction, a session-rename API endpoint, and streaming tool-progress frames.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.6.10 https://github.com/AtomicBot-ai/atomic-agent.git
# already have the repo? check out this version:
$ git checkout v0.6.10
```

Adds `image.generate` tool to draw pictures via OpenRouter, supporting square, landscape, portrait, or an exact aspect ratio, up to 4 images per call.

Adds `video.generate` tool to produce 4, 6, or 8 second clips from a text prompt or a seed image as the first frame; clips are saved as `video-<time>.mp4` and play inline in Telegram.

Adds `PATCH /api/sessions/{id}` endpoint with a `{ title }` body to rename a chat session; a manually set title is preserved and never overwritten by auto-naming.

Adds `os.fs.archive.list` and `os.fs.archive.extract` hints so the agent can open zip, tar, gz, and tgz attachments.

Adds new `paid_generation` approval category that gates both `image.generate` and `video.generate` behind an explicit user confirmation before spending money.

+2 more

WAS THIS USEFUL? 👍 👎

CopilotKit

Feedback ⓘ v1.80.0 NOTES ↗ RANK

CopilotKit connects AI agents to application interfaces with shared state, tools, and generative UI components.

> CopilotKit v1.80.0 adds reply ratings with comments to feed its Automatic Learning system, collapses repeated tool-call UI, and ships two breaking changes tightening MCP App widget sandboxing and A2UI catalog renderer prop types.

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

01 **A2UI catalog renderer prop types changed** BREAKING 80

`createCatalog` renderers (and the deprecated `createA2UICatalog` `render`) now receive props typed as the A2UI binder resolves them: Action props must be called as functions (e.g. props.onPress()), bound props are plain values (`props.level` is a `number`), the `{ path }` and `{ call }` shapes are gone, and a z.array(z.string()) prop now resolves to `any` .

— Gives exact before/after prop shapes migrating code must handle. v1.80.0

02 **MCP App widgets sandboxed with opaque origin** BREAKING 70

MCP App widgets now run in a sandbox without `allow-same-origin` , giving them an opaque origin. As a result `localStorage` , `sessionStorage` , `IndexedDB` , and cookies are no longer accessible to widgets, and `fetch` `/module-script` requests are sent with `Origin: null` .

— Explains exact mechanism and what access is now lost. v1.80.0

THINNER COVERAGE BELOW

03 **Reply ratings and comments for Automatic Learning** NEW 40

Adds reply ratings with comments collection in `@copilotkit/react-core` , feeding data into the Automatic Learning system.

— Names the package but not the API or UI surface used to rate. v1.80.0

04 **Collapsed consecutive tool-call UI** IMPROVED 35

Consecutive tool calls made in a row are now collapsed into a single UI element in `@copilotkit/react-core` , reducing repeated tool-call clutter in the chat UI.

— Describes the behaviour change but no config or toggle. v1.80.0

└──▷ **BREAKING ON UPGRADE**

! MCP App widgets now run in a sandbox without `allow-same-origin`, giving them an opaque origin; `localStorage`, `sessionStorage`, `IndexedDB`, and cookies are no longer accessible, and `fetch` `/module-script` requests go out with `Origin: null`.

! `createCatalog` renderers (and the deprecated `createA2UICatalog` `render`) now receive props typed as the A2UI binder resolves them: Action props must be called as functions (e.g. props.onPress()), bound props are plain values (`props.level` is a `number`), the `{ path }` and `{ call }` shapes are gone, and a z.array(z.string()) prop now resolves to `any`.

WAS THIS USEFUL? 👍 👎

openexecutive

Feedback ⓘ v0.6.5 NOTES ↗ RANK

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

> OpenExecutive v0.6.5 introduces per-channel safety levels that govern how outside content and own-mail handling are treated, with new UI guidance to configure them.

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

01 **Per-channel safety levels in orchestrator** NEW 45

The orchestrator now supports per-channel safety-level configuration, with automatic checking of all outside content against the configured level. The delegation module's own-mail hold now follows this configured safety level, and the UI adds an explanation of safety levels with a worked example to guide configuration.

— Describes mechanism and scope but no exact config keys or UI path v0.6.5

WAS THIS USEFUL? 👍 👎

suede-creator-skills

Feedback ⓘ v0.19.3 NOTES ↗ RANK

Open-source skills for Claude Code and Codex, with an MCP server, a PyPI package and Suede Awesome Everything, a directory that scores AI agent skills 0 to 99. Created by Jason Colapietro, founder and CEO of Suede AI.

> suede-creator-skills added five new offline auditing capabilities to the suede-seo-audit skill, all running without external API calls.

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

01 Offline SEO audit capabilities in suede-seo-audit NEW

62

The `suede-seo-audit` skill gained five new offline capabilities: hreflang relationship auditing (fully offline, no provider or API calls), image markup auditing, metadata snapshot and drift detection, opt-in `Product` / `Offer` schema checks, and deterministic SERP clustering from user-supplied SERP exports. All five are bundled as a self-contained Node 22+ CLI and modules inside the skill.

— Names five capabilities and runtime but no commands or config shown v0.19.3

WAS THIS USEFUL? 👍 👎

claw-orchestrator

Feedback ⓘ v7.14.0 NOTES ↗ RANK

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

> claw-orchestrator's biggest addition this window is native support for Agent Client Protocol agents as sessions, alongside JSON-scriptable session control, an unattended `goal` workflow node for Codex, a `launcher` config for sandboxing engine invocations, and two safety guards: a dirty-worktree check before autoloop starts and a block on OpenCode 2 at session start.

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

01 **ACP engine support for running any ACP agent** NEW 95

Adds `engine: "acp"` support with `acpAgent` (a preset id or inline `{ name, command, args, env }`) to run any Agent Client Protocol agent as a session over stdio, using `session/prompt`, `session/resume`, and `session/load`. Bundled presets `kimi-code` and `dsh` are included, and sessions are started via `clawo session-start -e acp --acp-agent <id>`.

Start an ACP session using the bundled `kimi-code` preset, then send a prompt and capture the full result as a single JSON line for downstream processing.

```
$ clawo session-start -e acp --acp-agent kimi-code
clawo session-send --json --message-file prompt.txt
```

— Names protocol methods, config shape, presets and exact start command. v7.14.0

02 **Goal RPCs restructured to app-server protocol** BREAKING 80

`codex_goal_set`, `codex_goal_get`, and `codex_goal_clear` now use the app-server's goal RPCs (`thread/goal/set`, `thread/goal/get`, `thread/goal/clear`) and return `{ ok, goal }` instead of sending `/goal …` as chat text; `codex_goal_set` now accepts `tokenBudget`, and goal status covers six states: `active`, `paused`, `blocked`, `usageLimited`, `budgetLimited`, and `complete`. The `timeout` parameter is removed from all three calls; callers must migrate to `tokenBudget` on `codex_goal_set` or drop the argument, and integrations parsing the old chat-text response will break.

— Full before/after and migration path, but no runnable example given. v7.13.0

03 **Dirty-worktree guard on `autoloop_start`** NEW 80

`autoloop_start` now enforces a clean Git worktree: staged, unstaged, and untracked changes — including submodule dirt hidden by repo config and untracked files hidden by `status.showUntrackedFiles=no` — cause the start to throw before any run directory or Planner session is created. A second-start exception allows resume when the only dirt is untracked (`??`) or index-clean worktree modifications (` M`) under `tasks/<id>/` while `tasks/<id>/goal.json` is a regular file.

— Precise mechanism and edge cases but no runnable command. v7.11.0

---

04 **`goal` workflow node and `clawo goal` command** NEW 75

Adds the `goal` workflow node, a `goal` template, and the `clawo goal` command to run a Codex `/goal` on a `codex-app` thread unattended, with checkpointed thread IDs and backoff retries up to `capacityRetry.max` ; progress is visible via `clawo workflow show` .

Inspect the status, token spend, and resume count of a running `goal` node without leaving the CLI.

```
$ clawo workflow show
```

— Names node, command, retry config key and inspection command. v7.13.0

---

05 **OpenCode 2 blocked at session start** BREAKING 75

claw-orchestrator now refuses OpenCode 2 (installed via `opencode upgrade --major`) at session start with a stated reason, because `run --format json` reports no tokens or cost and the `--variant` flag is absent. OpenCode 1.x and sessions whose `opencode --version` cannot be read are unaffected; users on OpenCode 2 must downgrade to OpenCode 1.x to continue.

— Clear cause and migration step, but no CLI example of the block itself. v7.11.0

---

06 **JSON output flags for scripted session control** NEW 70

Adds `--json` and `--message-file` flags to `clawo session-send` , and `--json` to `clawo session-status` , enabling scripts to drive sessions with machine-readable output and file-sourced messages.

Start an ACP session using the bundled `kimi-code` preset, then send a prompt and capture the full result as a single JSON line for downstream processing.

```
$ clawo session-start -e acp --acp-agent kimi-code
clawo session-send --json --message-file prompt.txt
```

Poll session status in a CI script by reading machine-readable JSON output rather than parsing human-formatted text.

```
$ clawo session-status --json
```

— Flags and commands named but no deeper mechanism given. v7.14.0

07 **`launcher` sandbox prefix for session config** NEW 65

Adds `launcher` session config to specify an argv prefix that wraps every engine invocation, enabling callers to run agents inside a custom sandbox.

Wrap every engine the session launches inside a sandbox binary using the new `launcher` config key.

```yaml
launcher:
  - /usr/local/bin/sandbox-exec
  - -p
  - my-sandbox-profile
```

— Config key named with a working YAML example, but mechanism is brief. v7.13.0

┗━━▷ BREAKING ON UPGRADE

! `codex_goal_set`, `codex_goal_get`, and `codex_goal_clear` no longer accept the `timeout` parameter; callers relying on `timeout` must migrate to `tokenBudget` (on `codex_goal_set`) or remove the argument.

! `codex_goal_set`, `codex_goal_get`, and `codex_goal_clear` now return `{ ok, goal }` instead of sending `/goal …` as chat text — integrations that parsed a chat-text response will break.

! OpenCode 2 (installed via `opencode upgrade --major`) is now refused at session start; users running OpenCode 2 must downgrade to OpenCode 1.x to continue using claw-orchestrator.

WAS THIS USEFUL? 👍 👎

octop-harness

Feedback ⓘ v1.0.5 NOTES ↗ RANK

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

> Across four releases, octop-harness added persistent MCP client sessions, user-maintained memory documents with write protection, an optional decision model to narrow tool/skill selection before the first token, and extensive memory auxiliary call tuning (timeouts, retries, token budgets, thinking-mode control), alongside new web search providers, channel-type plugins, PII detection, and Responses API support.

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

01 Memory auxiliary call configuration and tuning NEW

95

Adds a family of config keys to tune memory auxiliary calls independently from the main chat model: `memory_aux_max_tokens` sets a client-side token budget (separate from `memory_extract_max_tokens`); `memory_aux_extra_body` passes vendor-specific settings to OpenAI-compatible auxiliary endpoints (explicit values override `memory_aux_thinking` `-generated` keys); `memory_aux_max_retries` (default 2, range 0–10) and `memory_aux_retry_backoff_s` (default 1s, doubles up to 60s) control auxiliary retry behavior via an independent SDK instance; `memory_aux_light_timeout_s` and `memory_aux_heavy_timeout_s` set separate read timeouts for auxiliary calls; and `memory_aux_thinking` (`'auto'` or `'off'`) disables deep-thinking mode for known Qwen3/QwQ/DeepSeek-R/reasoner models on OpenAI-compatible endpoints, with `'on'` preserving the model default.

Cap token spend on memory auxiliary calls while keeping a separate extraction budget, and supply a vendor reasoning hint only to auxiliary calls.

```python
config = HarnessAgentConfig.from_env(
    workspace_dir=root,
    name="my-agent",
    memory_aux_max_tokens=1024,
    memory_extract_max_tokens=2048,
    memory_aux_extra_body={"reasoning_effort": "low"},
)
```

Configure independent timeouts and thinking-mode suppression for memory auxiliary calls when using a Qwen3 model on an OpenAI-compatible endpoint.

```python
config = HarnessAgentConfig.from_env(
    workspace_dir=root,
    name="my-agent",
    memory_aux_light_timeout_s=60,
    memory_aux_heavy_timeout_s=300,
    memory_aux_thinking="auto",
    memory_extract_max_tokens=2048,
)
```

— Every config key named with defaults, ranges, and two runnable examples. v1.0.3 v1.0.2

02 **memory_save integration with octop-memory** NEW 86

Adds memory_save(content, topic) to persist remember requests directly through octop-memory (requires octop-memory>=1.0.2); memory_search / memory_get retrieve additional history.

Persist a fact from a user remember request directly into octop-memory during an agent turn.

```python
memory_save("Alice prefers dark mode", topic="user_preferences")
```

— Named functions with version requirement and a runnable example. v1.0.4

03 **User-maintained memory documents with write protection** NEW 76

Introduces USER.md and MEMORY.md as user-maintained documents loaded once per session; agent file tools are blocked from writing to them, with write protection enforced by the existing FilesystemGuardMiddleware. The new manual_document_policy config key can override how the memory toggle controls read/write access on these files.

— Names files, middleware and config key but no runnable example. v1.0.4 v1.0.5

04 **Pause and drain background memory tasks** NEW 60

Adds MemoryMiddleware.pause_background method to let a host pause and drain all background memory tasks per expert, enabling safe backup and restore.

— Named method with purpose but no example of calling it. v1.0.3

THINNER COVERAGE BELOW

05 **Decision model for skill and tool filtering** NEW 56

Introduces an optional decision model that narrows the skills, tools, and teammates presented to the main model before its first token; falls back to the full catalogue on decision failure and passes recent conversation context to the scorer.

— Describes mechanism and fallback but no config key to enable it. v1.0.4

06 **Persistent MCP sessions via ManagedMCPClient** NEW 54

Adds `ManagedMCPClient` class for persistent MCP sessions, enabling long-lived MCP tool connections within an agent runtime.

— Names the class but gives no usage example or API detail. v1.0.5

07 **Channel-type support in plugin system** NEW 52

Adds channel-type support to the plugin system: plugins can register channel implementations and display metadata; first-registered channel wins on type conflicts.

— Explains conflict behavior but no API names or example. v1.0.4

08 **Per-model Chat Completions or Responses API selection** NEW 50

Adds support for selecting Chat Completions or Responses API per model on OpenAI-compatible endpoints, with optional explicit Responses → Chat Completions fallback.

— Describes the option but no config key or example given. v1.0.3

09 **New optional web search providers** NEW 40

Adds optional web search providers: you.com, Tencent Yuanbao SearchPro, and DeepSeek Search.

— Lists providers but no config key or setup steps shown. v1.0.5

10 **Virtual-path conversion for subtree experts** NEW 33

Adds virtual-path conversion for subtree experts, enabling explicit virtual paths to be resolved in multi-agent scenarios.

— Names the capability but no mechanism or usage shown. v1.0.2

11 **PII detection for China phone and ID numbers** NEW 30

Adds PII detection for mainland China mobile phone numbers and resident ID numbers.

— Bare statement with no mechanism or configuration detail. v1.0.2

WAS THIS USEFUL?  

OpenClaw

Feedback ⓘ v2026.10.7 NOTES ↗ CODE ↗ RANK

OpenClaw is a personal AI assistant platform that connects messaging channels, tools, and model providers through a gateway.

> OpenClaw v2026.10.7 adds support for the Gemini 4 Argon model.

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

01 **Gemini 4 Argon model support** NEW 35

OpenClaw now supports the Gemini 4 Argon model as a provider option.

— Names the model but no setup or config detail given. v2026.10.7

WAS THIS USEFUL? 👍 👎

agentlas-os

Feedback ⓘ v1.2.68 NOTES ↗ RANK

Agent OS: keep specialist agents in a hub, spin up a temporary orchestrator per task. Local-first, works with any model.

> This window centered on auth status reliability — automatic token renewal with a new `--no-refresh` flag and MCP exposure — alongside Ollama context-window fixes, a new long-running goal-tracking surface, model-led secret review on upload, and two breaking changes: removal of Gemini CLI as a supported host and migration of memory curation to curate().

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

01 **Automatic token renewal in auth status** NEW 90

`auth status` now automatically renews an expired access token on restart or app update, reporting `sign_in_required` when the grant is refused, `refresh_deferred` when the server is unreachable, and a descriptive error on unreadable records; the new `--no-refresh` flag skips this renewal and returns the plain stored record. The same renewal logic is now exposed as the `agentlas_auth_status` MCP tool.

Check sign-in state without triggering a token refresh — useful in scripts that should not side-effect credentials.

```
$ auth status --no-refresh
```

— Command, flag, MCP tool and error codes all named v1.2.68

02 **Gemini CLI removed as supported host** BREAKING 80

Gemini CLI is removed as a supported host; the installer and `hephaestus update` delete engine-owned files from `~/.gemini/extensions`, `~/.gemini/commands`, and `~/.gemini/hephaestus-extension-source` on upgrade.

— Exact file paths named, clear breaking impact v1.2.64

03 **Configurable runtime generation retention** NEW 75

The `AGENTLAS_RUNTIME_KEEP` environment variable (default `2`) controls how many runtime generations are retained on each install and update, now including previously-skipped hidden `.generations/` folders.

— Env var with default and scope, directly settable v1.2.65

04 **Reasoning-effort ceiling for /hep-orch** NEW 75

Adds a `role.effort=` parameter to `/hep-orch` to set a per-orchestration reasoning-effort ceiling, as in `/hep-orch role.effort=low`.

Cap the reasoning effort for a heavy orchestration run to control token spend.

```
$ /hep-orch role.effort=low
```

— Runnable command with parameter shown v1.2.64

---

05 **Memory curation moved to canonical model** BREAKING 75

Core automatic memory curation now runs through the canonical model: curate() queues legacy tickets, and `stop_hook` no longer admits learnings itself — callers relying on `stop_hook` to ingest learnings must migrate to curate().

— Names functions and migration path for a breaking change v1.2.64

---

06 **Ollama context window auto-sizing** NEW 70

The Ollama runner's context window is now sized from each model's trained maximum and Ollama is instructed not to truncate, preventing silent prompt cutoff on machines with a small default window. A new `local_context_limit_exceeded` error code is raised when a prompt exceeds that trained maximum.

— Names error code and mechanism but no direct user action v1.2.68

---

07 **Model-led secret review on upload** NEW 70

The connected or host model can only remove lines during secret review on upload and can never override a deterministic pattern hit; review receipts carry `file`, `line`, and `kind` fields only.

— Mechanism and receipt fields named, no direct command v1.2.64

---

08 **Context Map v4 captures CI/test support files** IMPROVED 65

The code-map policy advances to `v4`: Context Map now captures local verification dependencies — support modules and fixture data named by CI workflows and test commands — even when Git ignores them.

— Explains scope and version but no invocation path v1.2.67

---

09 **Storage location prompt for built agents** NEW 60

After build verification, Agentlas OS now prompts where to store a built agent — Cloud, public Hub, or local only — and registers built agents in the local roster automatically.

— Describes new UX flow but no command name v1.2.64

THINNER COVERAGE BELOW

10 **Descriptions in Hub search results** IMPROVED 55

Marketplace, owner Cloud, and bookmark search tools now return taglines for every search scope, giving Hub candidates their own descriptions so the local rerank can read more than names.

— Describes mechanism but no command or flag given v1.2.67

11 **Long-running goal tracking via /hep-goal** NEW 55

Adds a `/hep-goal` plan tree, a `goal.strategy` MCP tool, and a `hephaestus goal` CLI command for setting and tracking long-running, measurable goals.

— Named in summary only, no mechanism or usage detail v1.2.65

12 **Time budget for connected-model judgments** IMPROVED 50

Applies a five-minute time budget to connected-model judgments across the canonical runtime and the Claude/Codex adapters.

— States limit and scope but no configurable surface v1.2.63

└──▷ BREAKING ON UPGRADE

! Gemini CLI is removed as a supported host; the installer and `hephaestus update` delete engine-owned files from `~/.gemini/extensions`, `~/.gemini/commands`, and `~/.gemini/hephaestus-extension-source` on upgrade.

! `stop_hook` no longer admits learnings itself; callers relying on `stop_hook` to ingest learnings must migrate to curate().

WAS THIS USEFUL?  

PydanticAI

Feedback ⓘ v2.61.0 NOTES ↗

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

v2.61.0 adds structured citations to `TextPart`, makes the durable engine extension surface (`ParameterTransports`, `DurableRunContext`) public, adds `RealtimeModelProfile.supports_audio_output`, and bounds shell job logs with `Shell.max_log_bytes`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.61.0 https://github.com/pydantic/pydantic-ai.git
# already have the repo? check out this version:
$ git checkout v2.61.0
```

Adds structured citations to `TextPart`, with replay into message history and support in the Vercel AI adapter.

Makes the durable engine extension surface public: operation types, `ParameterTransports`, `DurableRunContext`, and run identity hooks.

Adds `RealtimeModelProfile.supports_audio_output` so text-only realtime models default to text output.

NEW SUPPORTED PLATFORM Adds `RealtimeClientSecret.webrtc_url`, carrying `webrtcfilter=on` on Azure OpenAI.

Bounds background and `shell` job logs with `Shell.max_log_bytes`.

+3 more

NEW INTEGRATION Citations normalized from Bedrock, OpenRouter, and xAI

NEW INTEGRATION Kimi model names added

WAS THIS USEFUL? 👍 👎

LiveKit Agents

Feedback ⓘ livekit-agents@1.8.9 NOTES ↗

RANK

LiveKit Agents builds realtime voice and video AI agents on LiveKit's WebRTC infrastructure, with turn detection, interruption handling, and telephony.

> LiveKit Agents 1.8.9 adds a provider-spanning DecisionModel for batched LLM evaluations, new OpenTelemetry attribute hooks and silent background-tool progress, five new speech/avatar plugin integrations, and several breaking changes to STT flushing, Rime, Gradium and Inworld behavior.

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

01 DecisionModel for batched LLM evaluations NEW

91

Adds `inference.DecisionModel` for evaluating named decisions using LiveKit credentials, batching evaluations and reporting refusals per decision without discarding other answers (also supporting `openai/gpt-6-luna-decisions`), `openai.DecisionModel` for OpenAI's Decisions API, and `microsoft_ai.DecisionModel` for batched decisions through Microsoft Foundry via `Microsoft-Decision-1`.

Evaluate a `handoff` decision using LiveKit Inference credentials instead of a provider API key, getting per-decision refusal reporting.

python

```python
model = inference.DecisionModel("openai/gpt-6-luna-decisions")
response = await model.evaluate(
    input=chat_ctx,
    decisions={"handoff": llm.Probability("The caller requests a human.")},
)
```

— Names three provider classes and a runnable usage example livekit-agents@1.8.9

02 Custom llm_request span attributes via gen_ai API NEW

80

Adds gen_ai.set_llm_request_attributes() to attach prompt metadata directly to the `llm_request` OpenTelemetry span from a custom `llm_node`, scoped to that node's calls, enabling tracing in tools like Langfuse without a custom OpenTelemetry processor.

Attach prompt metadata to the `llm_request` span from inside a custom `llm_node` for tracing in tools like Langfuse, without a custom OpenTelemetry processor.

```python
async def llm_node(self, chat_ctx, tools, model_settings):
    gen_ai.set_llm_request_attributes(
        {"langfuse.observation.prompt.name": "support-agent"}
    )
    async for chunk in Agent.default.llm_node(
        self, chat_ctx, tools, model_settings
    ):
        yield chunk
```

— Named function and span with a working code example `livekit-agents@1.8.9`

03 **Silent background-tool progress updates** NEW 65

Adds a `silent=True` parameter to ctx.update() to record background-tool progress without prompting a spoken response; final background-tool results can now trigger follow-up tool calls.

Report background-tool progress mid-call without triggering a spoken agent response.

```python
await ctx.update("checking the fare rules", silent=True)
```

— Named flag and example call, but scope of follow-up behavior is brief

`livekit-agents@1.8.9`

04 **New speech and avatar plugins** NEW 60

Adds ThunderPhone realtime speech-to-speech (including `bolt` mode) for saved-agent and inline sessions, AssemblyAI `assemblyai-tts-1` streaming TTS with word timestamps and six languages, Keyframe `persona-2.0-live` for speech-timed avatar expressions, Fish Audio `drama-3` Inference TTS with free-form expressive steering, and Speechify `simba-3.5` streaming synthesis with timestamps.

— Five named plugins and models but no usage example `livekit-agents@1.8.9`

05 **Manual flush support across STT engines** BREAKING 60

Adds manual flush support to ElevenLabs STT; supported engines now flush at manual and detected turn boundaries, so buffering custom nodes must disable `manual_flush`.

— Explains new behavior and migration note for custom nodes `livekit-agents@1.8.9`

THINNER COVERAGE BELOW

06 **Minimum dependency version bumps** BREAKING 55

OpenAI SDK minimum version is now `3.28+`; Phonic requires `phonic>=0.32.36`; framework requires `livekit-protocol>=1.1.28`.

— Names exact version floors for three dependencies `livekit-agents@1.8.9`

07 **LemonSlice avatar configuration arguments** IMPROVED 45

Adds `audio_sample_rate` and `model` arguments to LemonSlice avatar configuration.

— Named config keys but no usage shown `livekit-agents@1.8.9`

08 **Rime WebSocket v1 requires explicit model** BREAKING 45

Rime WebSocket v1 now requires an explicit `model` parameter; connections that relied on URL-path detection will break.

— Clear before/after but no migration example `livekit-agents@1.8.9`

09 **Gradium voice/voice_id conflict now errors** BREAKING 45

Gradium: passing both `voice` and `voice_id` now raises `ValueError`; callers must choose one.

— Names exact fields and exception but minimal context `livekit-agents@1.8.9`

10 **Inworld connection-pooling parameters deprecated** DEPRECATED 45

Inworld: `max_connections` and `idle_connection_timeout` are deprecated and ignored; TTS now uses one WebSocket per instance.

— Names deprecated params and new behavior, no migration path given `livekit-agents@1.8.9`

11 **Gemini gemini-3.8-flash default chat model** NEW 40

Adds Google `gemini-3.8-flash` to supported chat models, selected as the new default.

— Names model but no configuration detail `livekit-agents@1.8.9`

└──▷ BREAKING ON UPGRADE

! Rime WebSocket v1 now requires an explicit `model` parameter; connections that relied on URL-path detection will break.

! Gradium: passing both `voice` and `voice_id` now raises `ValueError`; callers must choose one.

! Inworld: `max_connections` and `idle_connection_timeout` are deprecated and ignored; TTS now uses one WebSocket per instance.

! STT flushing: supported engines now flush at manual and detected turn boundaries; buffering custom nodes must disable `manual_flush`.

! OpenAI SDK minimum version is now 3.28+; Phonic requires `phonic>=0.32.36`; framework requires `livekit-protocol>=1.1.28`.

WAS THIS USEFUL?  

Daily Pipecat

Feedback ⓘ v1.14.0 NOTES ↗ RANK

Pipecat is a framework for realtime voice and multimodal agents, orchestrating speech-to-text, model inference, and text-to-speech as a streaming pipeline.

> Pipecat v1.14.0 adds three new LLM services (Perplexity Agent, NVIDIA Omni, Smallest AI), a Zoom Scribe Live STT integration, and an LLM-output text-transform pipeline, alongside NVIDIA STT speaker diarization and a batch of breaking default-model changes across Vertex, NVIDIA, Speechify and Rime.

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

01 **Default model version bumps across LLM/TTS services** BREAKING 95

`GoogleVertexLLMService` default model changed from `gemini-3.6-flash` to `gemini-3.8-flash` (pass settings=GoogleVertexLLMService.Settings(model='gemini-3.6-flash') to keep the previous model); `NvidiaLLMService` default model changed to `nvidia/nemotron-3.5-super-vl-120b-a12b` (set settings=NvidiaLLMService.Settings(model='nvidia/nemotron-3-super-120b-a12b') to keep the previous model); `SpeechifyHttpTTSService` default model changed to `simba-3.5` (set `settings.model` to `simba-3.2` or `simba-3.0` to keep the previous behaviour); and `RimeV1TTSService` now requires an explicit `settings.model`, since the previous implicit default no longer applies.

— Every service, old/new default and migration override named. v1.14.0

02 **LLM output text-transform pipeline** NEW 88

Adds `text_transforms` to `LLMTextProcessor` with add_text_transformer() and remove_text_transformer(), so registered transforms rewrite LLM output before it splits toward TTS, context, and client; the callable type is `TextTransform` in `pipecat.utils.text.transforms`.

Strip unwanted emoji and stage directions from LLM output before they reach TTS, context, and the client.

```python
from pipecat.utils.text.transforms import TextTransform
import re

def remove_emoji(text: str) -> str:
    return re.sub(r'[\U00010000-\U0010ffff]', '', text,
flags=re.UNICODE)

llm_text_processor.add_text_transformer(('*', remove_emoji))
```

— Runnable example demonstrates the exact API. v1.14.0

03 **New LLM service integrations: Perplexity, NVIDIA Omni, Smallest**

NEW 85

Adds `PerplexityAgentHttpLLMService` for Perplexity's Agent API with web search and function calling, where `Settings` exposes `preset`, `web_search`, and `max_steps`, defaulting to `openai/gpt-5.6-luna` with web search on; adds `NvidiaOmniLLMService` for NVIDIA Nemotron Omni multimodal models accepting speech, images, video, and text, with an `audio_input` constructor argument (default `True`) that replaces the STT stage, plus run_multimodal_inference() for one-off requests outside the conversation; and adds `SmallestLLMService` connecting to Smallest AI's Electron model via the OpenAI-compatible Waves API, offered as `Smallest` in `pipecat init`.

Use Perplexity's Agent API with web search enabled and a step limit in a voice pipeline.

```python
from pipecat.services.perplexity import
PerplexityAgentHttpLLMService

llm = PerplexityAgentHttpLLMService(
    api_key=os.environ['PERPLEXITY_API_KEY'],
    settings=PerplexityAgentHttpLLMService.Settings(
        model='openai/gpt-5.6-luna',
        web_search=True,
        max_steps=5,
    )
)
```

— Mechanism, settings and defaults named for all three; one runnable example. v1.14.0

04 **Zoom AI streaming STT integration** NEW 85

Adds `ZoomAISTTService` for Zoom Scribe Live streaming STT, installable via `pipecat-ai[zoom-ai]`. `Settings.silence_duration_ms` (default 350) controls turn end, and

`Settings.diarization` enables speaker labels in `TranscriptionFrame.result`.

– Install path and settings named, no code example. v1.14.0

05 **Twilio API-key auth and region configuration** IMPROVED

80

Adds `api_key_sid` and `api_key_secret` to `TwilioFrameSerializer` so auto hang-up authenticates with a Twilio API key instead of the account Auth Token, and the development runner's Twilio serializer now reads `TWILIO_API_KEY`, `TWILIO_API_SECRET`, `TWILIO_REGION`, and `TWILIO_EDGE` environment variables for non-US1 region support.

– Exact fields and env vars named, no code sample. v1.14.0

06 **NVIDIA STT speaker diarization** NEW

78

Exposes speaker diarization in `NvidiaSTTService` transcription frames; Settings(speaker_diarization=True, speaker_format='Speaker {speaker}: {text}') labels each contiguous speaker run as its own frame with the NVIDIA speaker tag included in `user_id`.

– Settings and output format named, no runnable example. v1.14.0

07 **strip_markdown strips URLs by default** BREAKING

75

`strip_markdown` now strips URLs from Markdown links and images by default; register StripMarkdown(strip_urls=False) or pass `strip_urls=False` to `VoiceFormatter` to restore the old behaviour.

– Exact flag and restoration path given. v1.14.0

08 **Speechmatics STT segmentation and sample-rate support** IMPROVED

70

Adds `emit_sentences` to `SpeechmaticsSTTService.Settings`; when enabled, Speechmatics closes a segment at every sentence boundary so each sentence is emitted as its own final transcription. `SpeechmaticsSTTService` also now supports 8 kHz audio input alongside 16 kHz.

– Named setting and sample rates, no usage example. v1.14.0

09 **pipecat context-hub repos command** NEW

60

Adds the `pipecat context-hub repos` command via the `pipecat-ai-context-hub` 0.9.0 floor bump, enabling adding and removing indexed repos and letting hub queries wait out a running `refresh`.

– Named command, no invocation example shown. v1.14.0

THINNER COVERAGE BELOW

10 **LemonSlice transport model and sample-rate options** IMPROVED

57

`LemonSliceTransport` now accepts a `model` parameter to specify LemonSlice models and an `audio_sample_rate` parameter to set audio input/output sample rate (defaulting to 24 kHz).

— Parameters and default named, no example. `v1.14.0`

---

11 **RTVI bot-output type field** NEW

50

Adds `output_type` field to the RTVI `bot-output` message, indicating whether the text is a sentence, word, backchannel, or a custom type such as `'code'`.

— Field documented but no usage context. `v1.14.0`

╰──▷ BREAKING ON UPGRADE

! `RimeV1TTSService` now requires an explicit `settings.model`; the previous implicit default no longer applies.

! `GoogleVertexLLMService` default model changed from `gemini-3.6-flash` to `gemini-3.8-flash`; pass `settings=GoogleVertexLLMService.Settings` (model='gemini-3.6-flash') to keep the previous model.

! `NvidiaLLMService` default model changed to `nvidia/nemotron-3.5-super-vl-120b-a12b`; set `settings=NvidiaLLMService.Settings` (model='nvidia/nemotron-3-super-120b-a12b') to keep the previous model.

! `SpeechifyHttpTTSService` default model changed to `simba-3.5`; set `settings.model` to `simba-3.2` or `simba-3.0` to keep the previous behaviour.

! `strip_markdown` now strips URLs from Markdown links and images by default; register StripMarkdown(strip_urls=False) or pass `strip_urls=False` to `VoiceFormatter` to restore the old behaviour.

WAS THIS USEFUL? 👍 👎

Vercel AI SDK

Feedback ⓘ ai@5.0.281 NOTES ↗ RANK

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

> Vercel AI SDK shipped stricter tool-choice enforcement, an experimental streaming speech API, and a new `interrupted` finish reason for truncated provider streams — the latter being a breaking change for exhaustive `FinishReason` handling.

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

01 interrupted finish reason for truncated streams BREAKING

95

Adds `interrupted` as a new `FinishReason` value for provider streams that end without a completion signal, preserving partial output without raising an error while halting tool execution, agent continuation, and automatic UI follow-ups. Applications should check `finishReason === 'interrupted'` to detect incomplete responses; this breaks exhaustive switches (e.g. with an `assertNever` fallback) and exhaustive mappings like `Record<FinishReason, string>`, which must be updated to handle the new value.

Detect an interrupted (truncated) stream response and handle it separately from a clean finish, rather than treating it as an error.

typescript

```typescript
import { generateText } from 'ai';

const { text, finishReason } = await generateText({
  model: 'openai/gpt-6-astra',
  prompt: 'Write a very long story.',
});

if (finishReason === 'interrupted') {
  console.warn('Stream ended without a completion signal; output may be partial:', text);
} else {
  console.log('Complete response:', text);
}
```

— Exact field, check condition, runnable example, and migration impact noted ai@7.0.149

02 ToolChoiceViolationError for toolChoice: 'none' NEW

90

Introduces `ToolChoiceViolationError`, thrown in `generateText` and `streamText` when a model returns tool calls despite `toolChoice: 'none'`, preventing local execution and withholding streamed tool parts from client-side tools. The constraint is also forwarded to the Hugging Face Responses API so it is enforced at the provider level.

Catch a `ToolChoiceViolationError` when a model ignores `toolChoice: 'none'` and returns tool calls anyway — useful for strict enforcement in production pipelines.

typescript

```
import { generateText, ToolChoiceViolationError } from 'ai';

try {
  const { text } = await generateText({
    model: 'openai/gpt-6-astra',
    toolChoice: 'none',
    tools: { /* ... */ },
    prompt: 'Summarise this document without using any tools.',
  });
} catch (err) {
  if (err instanceof ToolChoiceViolationError) {
    console.error('Model attempted a tool call despite
toolChoice: none', err);
  }
}
```

— Named error class, exact APIs, and a runnable try/catch example `ai@5.0.281`

03 **experimental_streamSpeech for streaming audio** NEW 75

Adds `experimental_streamSpeech` for streaming audio from text or text-delta streams using speech models, exporting new types `Experimental_SpeechTextPart`, `Experimental_SpeechStreamPart`, `Experimental_SpeechStreamUsage`, and `Experimental_StreamSpeechResult` from the `ai` package, plus `Experimental_MockStreamingSpeechModelV4` in `ai/test` for unit testing. Provider `error` parts are non-fatal and routed through an `onError` callback (default `console.error`); an abort or cancel rejects all result promises immediately, and `NoSpeechGeneratedError` gained an optional `cause` field.

— Names every export and config surface but no usage example given `ai@7.0.150`

└──▷ BREAKING ON UPGRADE

! Adding `interrupted` to the public `FinishReason` type breaks exhaustive switches (e.g. those using an `assertNever` fallback) and exhaustive mappings such as `Record<FinishReason, string>` — all must be updated to handle the new value.

WAS THIS USEFUL? 👍 👎

busabase

Feedback ⓘ v0.99.0 NOTES ↗ RANK

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

> busabase v0.99.0 adds email delivery via SMTP/SES with durable reply tracking.

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

01 **SMTP/SES email delivery with reply tracking** NEW 40

busabase can now send email through SMTP or SES, with durable tracking of replies to sent messages.

— Describes capability but no config keys or setup steps given v0.99.0

WAS THIS USEFUL? 👍 👎

Google ADK

Feedback ⓘ v2.13.0 NOTES ↗ RANK

The Agent Development Kit is Google's code-first Python toolkit for building, evaluating, and deploying agents, with built-in tools, sessions, and deployment to Vertex AI.

> Google ADK v2.13.0 adds a new classifier subsystem for workflow routing, full MongoDB support for sessions, long-term memory and toolset embedding, barge-in voice interactions for CascadeLive, and a breaking change switching telemetry to OpenTelemetry GenAI v2 by default.

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

01 OpenTelemetry GenAI v2 becomes default telemetry schema

85

BREAKING

Telemetry now defaults to OpenTelemetry GenAI v2 (`otel_semconv_1_44`), removing the `call_llm` span; set `ADK_TELEMETRY_SCHEMA_VERSION_OPT_IN=otel_semconv_1_36` to keep the legacy v1 schema.

— Exact schema names and migration env var given v2.13.0

02 TypesafeClassifier and classifier subsystem for routing NEW

80

Adds `TypesafeClassifier` , `ClassifierNode` , and `LlmClassifier` as a new classifier subsystem for workflow routing. `TypesafeClassifier` / `ClassifierNode` use TypeSafe's Jev models to return calibrated probabilities in a single forward pass without an LLM call, while `LlmClassifier` provides LLM-based classification.

— Names three new classes and the underlying model type v2.13.0

03 MongoDbToolset embedding and packaging improvements IMPROVED

70

Improves `MongoDbToolset` by adding Atlas Automated Embedding support so search queries are embedded server-side without Google embedding credentials, and by making it picklable for Agent Engine deployment, keeping MongoDB credentials out of Agent Engine packages.

— Two concrete, well-described mechanism changes v2.13.0

04 (new) MongoDB-backed session and memory services NEW

65

Adds `MongoDbSessionService` to persist sessions in MongoDB and share them across runner replicas, and `MongoDbMemoryService` for long-term memory via keyword match

on any MongoDB deployment, with no vector index or Google embedding call needed.

— Named class and effect, no setup detail Explains mechanism and requirement removed

v2.13.0

05 **Barge-in and speech-to-text/TTS for CascadeLive** NEW

60

Adds barge-in support to `CascadeLiveConnection` for real-time voice interactions, plus Gemini speech-to-text and text-to-speech for `CascadeLive`.

— Names classes and capability but no usage steps v2.13.0

THINNER COVERAGE BELOW

06 **Environment variable to suppress experimental warnings** NEW

55

Adds `ADK_SUPPRESS_EXPERIMENTAL_FEATURE_WARNINGS` environment variable to silence experimental feature warnings.

— Exact env var name is directly actionable v2.13.0

07 **Visual frame pack/unpack for live video** NEW

40

Adds `ZIP_STORED` visual frame pack and unpack helpers for live video.

— Bare addition with named format, no mechanism v2.13.0

08 **delete_memory API on BaseMemoryService** NEW

40

Adds `delete_memory` API to `BaseMemoryService`.

— Named API, no further detail given v2.13.0

09 **OpenSandbox and AgentEngineEnvironment for VMaaS deployments** NEW

40

Adds `OpenSandbox` environment and `AgentEngineEnvironment` for VMaaS deployments.

— Two named environments, no usage or config detail v2.13.0

10 **ForceEscalationPlugin for deterministic model consultation** NEW

35

Adds `ForceEscalationPlugin` for deterministic model consultation.

— Named plugin, purpose stated only briefly v2.13.0

11 **Session and turn trace views in evaluation plugins** NEW

30

Adds agent evaluation session and turn trace views in plugins.

— Generic description, no named surface v2.13.0

└──▷ BREAKING ON UPGRADE

! Telemetry now defaults to OpenTelemetry GenAI v2 (`otel_semconv_1_44`), removing the `call_llm` span; set `ADK_TELEMETRY_SCHEMA_VERSION_OPT_IN=otel_semconv_1_36` to keep the legacy v1 schema.

WAS THIS USEFUL? 👍 👎

CrewAI

Feedback ⓘ 1.15.29 NOTES ↗ RANK

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

Adds `call:` tool for direct app actions, offline guardrail evaluation runner, LiteLLM kwargs support, richer `crewai eval` commands, Flow tracing, notebook/JSONL loaders, and stable `AgentExecutor` .

└──▷ GET THIS VERSION ▾

```
$ git clone --branch 1.15.29 https://github.com/crewAIInc/crewAI.git
# already have the repo? check out this version:
$ git checkout 1.15.29
```

Adds `call:` tool to run integration app actions directly from an agent.

Adds offline JEV routing and guardrail evaluation runner.

Enables `crewai eval` to run the crew itself and return a verdict from a single command.

Enables `crewai eval` to export evaluations as files.

Adds support for LiteLLM-compatible kwargs on LLM calls.

+6 more

NEW INTEGRATION Azure DocumentDB vector search integration

WAS THIS USEFUL? 👍 👎

LangChain Deep Agents

Feedback ⓘ deepagents==0.7.31

NOTES ↗ RANK

Deep Agents provides an agent harness with planning, filesystem tools, subagents, and context management.

> LangChain Deep Agents 0.7.31 adds visibility into execution output retention within tracing.

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

01 **Retention metadata exposed in tracing** IMPROVED 30

Deep Agents now exposes execution output retention metadata in tracing, making retention signals visible to observability pipelines.

— No specific field names, APIs, or config keys given deepagents==0.7.31

WAS THIS USEFUL? 👍 👎

cotal

Feedback ⓘ v0.100.0 NOTES ↗ LATEST STABLE

The open standard for agent coordination

cotal v0.100.0 improves CLI diagnostics, credential routing, color control, and error transparency across run, channel, and backup commands.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.100.0 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.100.0
```

└──▷ TRY IT

Diagnose a failing endpoint call by seeing the full underlying error chain, not just the top-level code.

```
$ COTAL_DEBUG=1 cotal ep describe --creds caller.creds.json
```

Recover from a Ctrl-C-interrupted backup by taking over the orphaned create attempt once the original process is gone.

```
$ cotal backup recover backup-<uuid> --yes
```

Release a broker or daemon stuck in `stopping` state after a killed `cotal down`, once the dead process has exited.

```
$ cotal down --abandon-stop <request-id> --yes
```

Adds `COTAL_DEBUG=1` stack-trace output for failed commands — prints the failure stack and each causing error's stack, making underlying errors like timeout messages visible without changing the JSON envelope.

Adds `ai.cotal.ep.no-live-instances` detail code on the service registry's empty answer so callers can distinguish `no live manager` from an unreadable registry.

Adds `--abandon-stop <request-id> --yes` to `cotal down` to take a stop left by a killed or interrupted `down` process once that process is confirmed gone.

`cotal run show` and `cotal run journal` now surface per-step `errorMessage` alongside `errorCode` in `--json` output and list failures under a `Step errors` section in

the text view.

`config`, `config get`, and `status` now emit `PREFERENCE_INVALID` warnings when environment variables such as `COTAL_DETACH_KEY`, `COTAL_DEFAULT_AGENT`, or `COTAL_DEFAULT_PERSONA` hold values that `config set` would refuse, naming the variable and the reason.

+5 more

└──▷ **BREAKING ON UPGRADE**

! `ADMITTED_NODES`, `STRUCTURAL_NODES`, `RESERVED_NAMES`, `FORBIDDEN_GLOBALS`, and `MUTATING_METHODS` are now frozen arrays instead of `Set`s, and `CATALOG` is frozen — any code that called .has() or `.size` on these exports or assigned a `CATALOG` title will break; use .includes() and `.length` instead.

! Caller intent files written by `cotal auth mint` before this release are refused with `SCHEMA_INVALID` on `ep describe`, `ep invoke`, and `invoke` — re-issue with `cotal auth mint <name> --endpoint <class> --out <file>`.

▾ **28 more releases in this issue** · 2026-10-10 → 2026-10-23

v0.99.0 NOTES ↗ STABLE

Cotal v0.99.0 moves telemetry state to `~/.cotal/telemetry.json` and adds a per-person `account` hash to deduplicate multi-machine users.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.99.0 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.99.0
```

└──▷ **TRY IT**

Verify which account hash cotal will report in telemetry after signing in, to confirm deduplication is working across machines.

```
$ cotal telemetry
```

Moves the telemetry install-id file to `~/.cotal/telemetry.json` (or `%LOCALAPPDATA%\Cotal` on Windows); `COTAL_HOME` no longer redirects it, so one person's workspaces share a single install id.

Adds an `account` field to `command` and `install` telemetry events at the `regular` level: a 32-hex-character sha256 hash of the IdP URL and subject id of the cached sign-

in, enabling one person on multiple machines to count once; the field is `null` when no sign-in is cached.

Adds `cotal telemetry` output of the `account` hash, letting users inspect what identity token is reported.

`cotal attach` now keeps reconnecting when the user-auth service does not answer its base bearer exchange, matching the existing retry behaviour for the manager-caller exchange and an unreachable broker.

└──▷ **BREAKING ON UPGRADE**

! The telemetry level stored in a `COTAL_HOME` file is no longer read; run `cotal telemetry <level>` to set it again.

v0.98.0 NOTES ↗ STABLE

Adds `--daemon-home <dir>` to run isolated parallel daemons with their own session, identity, and socket.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.98.0 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.98.0
```

└──▷ **TRY IT**

Run a second isolated Cotal daemon — with its own session and identity — alongside the default, useful for managing separate spaces or tenants on the same host.

```
$ mkdir -m 700 /var/run/cotal-secondary
cotal daemon serve --daemon-home /var/run/cotal-secondary
```

Adds `--daemon-home <dir>` to `cotal daemon … serve` to run a separate daemon whose state — sign-in session, per-space identity, register request, service store, socket at `$XDG_RUNTIME_DIR/cotal/home-<hash>/daemon.sock`, and lifetime lock at `<dir>/daemon.lock` — lives entirely in that directory, independent of the default daemon.

A whitespace-only `COTAL_SECRET_STORE` environment variable is now refused at startup (when the manager or delivery daemon is constructed) with a message that names the variable, rather than passing silently and failing later inside the challenge.

v0.97.1 NOTES ↗ STABLE

Adds `--request-id <hex32>` to `cotal service-endpoint approve|refuse|name-bind` for reliable replay of recorded decisions.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.97.1 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.97.1
```

└──▷ TRY IT

Re-send a service-endpoint approval whose answer was lost in transit, using the request id printed by the original invocation.

```
$ cotal service-endpoint approve --request-id <hex32>
```

`cotal service-endpoint approve|refuse|name-bind` now prints the request id it was sent under, so a lost answer can be identified and replayed.

v0.97.0 NOTES ↗ STABLE

cotal v0.97.0 adds guest-side serve grant seam, CLI service-endpoint hosting, and own-broker caller issuance with retained signing seed.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.97.0 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.97.0
```

└──▷ TRY IT

Stand up a local auth service with full service-endpoint hosting — engine, public authority door, and authenticated host route — so services can register and guests can verify through it.

```
$ cotal up --user-auth --service-endpoints
```

Rotate the system-account signing seed so own-broker caller issuance picks up the new seed on next `cotal up --service-endpoints`.

```
$ cotal up --rotate-sys
```

Adds `authorizeGuestServeGrant`, the guest-side serve grant seam (SPEC §13.6), which turns a parsed `credential` result and embedded contract closure into a process-local serve grant consumed by `serveEndpoint` with no store access, and adds credential() to `ServiceEndpointRenewal`, returning a `ServiceEndpointCredential` (witness, host-authorized snapshot, and access bearer) for handoff to `authorizeGuestServeGrant`; concurrent bearer() and credential() calls share one request under the same persisted activate and renew ids.

The service authority's `GET` answer now includes `issuer` and `keys` fields (public Ed25519 coordinates matching `GET /jwks`), which `parseServiceAuthorityInfo` reads as a closed bounded record and serveAuthenticityOf(info) uses to build the seam's authenticity; published Ed25519 keys are now validated as canonical full-order points (in `parseServiceAuthorityInfo` and in the guest seam), and the door refuses to publish more than 16 keys.

Host service endpoints from the stock CLI via `cotal up --user-auth --service-endpoints`, which opens the service endpoint engine, the public `/service-endpoint-authority` door, and the new authenticated `/service-endpoint-host` route (without the flag both routes answer 404), and adds `cotal service-endpoint approve|refuse|proposal|operation|name|name-bind` subcommands to drive the `/service-endpoint-host` route as the signed-in user.

Own-broker service caller issuance: `cotal up --rotate-sys` retains the system-account signing seed in the root's secret store; `@cotal-ai/workspace` exports `retainSystemSigningSeed`, `readSystemSigningSeed`, and `systemSigningSeedKey`.

The service door now admits schema closures referenced by a registration's commands (`inputDigest` and `outputDigest`), publishing them with cluster artifacts so a door-registered service publishes its schemas the same way an in-process registrant does.

+1 more

└─▷ **BREAKING ON UPGRADE**

! credential() is now a required member of the `ServiceEndpointRenewal` interface — out-of-tree implementers must add it or their build will fail.

v0.96.1 NOTES ↗ STABLE

Adds `cotal daemon service-connect`, `service-disconnect`, and `service-status` on Linux for managed daemon-to-space connections.

└─▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.96.1 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.96.1
```

└──▷ **TRY IT**

Connect the daemon to a space as a managed service and confirm the broker accepted it before proceeding with automated workflows.

```
$ cotal daemon service-connect
cotal daemon service-status
```

Gracefully detach the daemon from the space — for example before a planned maintenance window — without stopping the daemon process itself.

```
$ cotal daemon service-disconnect
```

v0.96.0 NOTES ↗ STABLE

Adds `AuthProvider.serviceEndpointRenewal`, a self-renewing bearer token source for long-lived service endpoints.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.96.0 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.96.0
```

Adds `AuthProvider.serviceEndpointRenewal`, a renewal source that activates on first pull and auto-renews on subsequent pulls by replaying one persisted request per operation, exposing tokens only through bearer() and reporting outages as `unavailable` for retry.

v0.95.2 NOTES ↗ STABLE

Cotal v0.95.2 sends masked error text in usage telemetry on CLI failures.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.95.2 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.95.2
```

v0.95.1 NOTES ↗ STABLE

cotal v0.95.1 adds CLI-initiated daemon registration with owner approval flow.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.95.1 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.95.1
```

Adds daemon registration request via the CLI, requiring owner approval before the daemon connects.

v0.95.0 NOTES ↗ STABLE

Cotal v0.95.0 adds opt-out anonymous usage telemetry to the CLI.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.95.0 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.95.0
```

v0.94.7 NOTES ↗ STABLE

Cotal v0.94.7 adds `--abandon-stop` to gracefully exit stalled manager stops and preflight daemon connection checks.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.94.7 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.94.7
```

└──▷ TRY IT

Force a clean exit when a controller stack's manager stop is stalled waiting for a receipt that never arrives.

```
$ cotal down --abandon-stop
```

Adds `--abandon-stop` flag to `down` command to abandon a manager stop when no receipt settles, preventing indefinite hangs.

Adds daemon connection prerequisite checks to the CLI so misconfigured connections are caught before commands run.

v0.94.5 NOTES ↗ STABLE

Cotal v0.94.5 adds device-approval sign-in for the local daemon.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.94.5 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.94.5
```

v0.94.3 NOTES ↗ STABLE

Cotal v0.94.3 lets callers cancel device sign-in mid-flow.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.94.3 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.94.3
```

Adds the ability to cancel an in-progress device sign-in flow, so callers are no longer blocked waiting for authentication to complete or time out.

v0.94.1 NOTES ↗ STABLE

cotal v0.94.1 adds a local daemon status endpoint on Linux.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.94.1 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.94.1
```

v0.94.0 NOTES ↗ STABLE

Cotal v0.94.0 issues endpoint caller rows from native caller policy in the auth system.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.94.0 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.94.0
```

└──▷ BREAKING ON UPGRADE

! The `auth` endpoint caller row issuance now derives from native caller policy (marked as a breaking change in PR #4324); any integration relying on the previous caller-row behaviour may need review.

v0.93.0 NOTES ↗ STABLE

Cotal v0.93.0 requires approved app permissions for connectors and adds native delivery support.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.93.0 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.93.0
```

└──▷ BREAKING ON UPGRADE

! Connector setup now requires approved app permissions — existing connectors without approved permissions will break on upgrade.

v0.92.0 NOTES ↗ STABLE

v0.92.0 adds authenticated workflow cancellation, permissioned host vitals, caller-bound workflow submission, and native Claude message previews.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.92.0 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.92.0
```

Adds authenticated cooperative workflow cancellation to the runtime, enabling callers to cancel in-flight workflows with verified identity.

Adds permissioned host vitals to the manager, gating access to host health data by caller permissions.

Adds caller-bound keyed workflow submission to the manager, tying submitted workflows to a specific caller identity.

Hosts approved service endpoints in user-auth spaces, scoping service authority to authenticated user spaces.

Exports feedback submission from the connector-core.

+2 more

v0.91.0 NOTES ↗ STABLE

Cotal v0.91.0 adds `cotal chat`, `cotal config`, `cotal history`, `cotal docs`, and a unified completion status command with structured JSON/plain output throughout.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.91.0 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.91.0
```

└──▷ TRY IT

Join a live chat session on a specific channel as a named participant, outputting NDJSON for programmatic consumption.

```
$ cotal chat #incidents --stdio --name 'soc-bot'
```

Audit recent DM and channel history across a space, filtering to messages in the last hour, then stream new messages as they arrive.

```
$ cotal history #alerts --since 1h --dms --follow --json
```

Pin a default agent harness for a specific space so all spawns in that space use it without an explicit flag.

```
$ cotal config set harness my-hardened-harness --space prod-space
```

Adds `cotal chat [<channel>]` for live terminal sessions with `--plain` line output or `--stdio` NDJSON participant protocol, resolving auth mode automatically (open mesh, static-auth, or user-auth via `cli` grant), plus a `--name` flag to set the session display label (defaults to OS login and host with a short suffix).

Adds `cotal config` with subcommands `get <key>`, `set`, and `unset` for personal preferences: `harness`, `persona`, `detach-key`, `color`, and `editor` (with `--space` for per-space overrides of `harness` and `persona`), plus `--json` and `--plain` output on `config get` and bare `config`, resolving each key through an explicit flag, persona pin, named env var (`COTAL_DEFAULT_AGENT`, `COTAL_DEFAULT_PERSONA`, `COTAL_DETACH_KEY`, `FORCE_COLOR` / `NO_COLOR`, `VISUAL` / `EDITOR`), space value, personal value, then default; preferences are stored in `preferences.json` in the configuration directory, written atomically under a lock with `CONFLICT` protection on concurrent edits.

Adds `cotal history [<channel>]` to read retained messages with `--limit` (default 50, max 1000), `--since` (time, age, or stream sequence), `--from`, `--grep`, and `--dms` filters, plus `--follow` for live streaming, and adds `history clear` to purge all history in the selected space, with a confirmation prompt or `--yes` for scripted use.

Exposes `CotalEndpoint.multiChannelHistory` (now returning each message's stream sequence), chatRetention(), and channelReplayPolicy(channel) on the `CotalEndpoint` API.

Adds `cotal docs [<page>]` for offline protocol and CLI documentation: bare `docs` lists pages, a page name prints its Markdown, `--search <query>` shows best-matching sections, and `--refresh` fetches the version-pinned copy from `docs.cotal.ai`; also exports `docsIndex`, `readDocsPage`, `searchDocs`, and `capSection` from `@cotal-ai/connector-core` via a new `./docs` subpath.

+5 more

└──▷ BREAKING ON UPGRADE

- ! `-v` no longer prints the version and is refused; it is reserved for `--verbose`. Use `cotal version`, `--version`, or `-V` instead.
- ! `cotal completion <shell>` is replaced by `cotal completion script <shell>`. The old form exits 1 with nothing on stdout, so a hand-written source <(cotal completion bash) now loads nothing and names its replacement on stderr.

v0.90.0 NOTES ↗ STABLE

Cotal v0.90.0 adds internal service-authority contracts, revocation custody, and a dormant native service custody writer.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.90.0 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.90.0
```

States in the `cotal_run` tool's `page` description that a manager from release 0.88.0 or earlier refuses page mode.

Adds immutable local scoped-Describe receive context, complete-envelope fingerprints, and an explicit bounded read-only service Describe client contract, including closed service-authority contracts, provider HTTP client hooks, host control types, registration-bound access and renewal primitives, native service gate approval guards, credential-root access checks, native caller-policy storage, and create-only application-service name bindings.

Adds host-only fresh revocation custody and sealed finite leader snapshots for private tool-result prerequisites.

Adds a package-internal dormant writer for native service custody bootstrap and endpoint-first cancellation.

Adds a host-only service-callout mint seam with principal attribution, a separate client inbox nonce, and an original fixed expiry.

+2 more

v0.89.0 NOTES ↗ STABLE

Cotal 0.89.0 adds scoped account connection census, bounded run inventory, and bearer-expiry enforcement keyed to IdP proof `exp`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.89.0 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.89.0
```

Bounds exchanged bearers by the IdP proof's absolute `exp` at signing time, carrying the proof's `exp` and the `session-caller` session's `exp` through the credential hook so the issuer signs the earlier of `now + ttl` and that ceiling, and refuses once it has passed — closing a window where a slow `mintConnectCredential` hook could push a bearer or NATS user JWT past the proof's expiry.

Adds a host-only, short-lived account census credential that grants exactly one account's `CONNZ` request and account-qualified replies, without eviction rights or terminal-retirement claims.

Adds explicit bounded diagnostic pages for native run inventory, letting CLI and MCP callers continue a pinned scan without a full-list reply or an absence claim, while preserving legacy list requests and existing permissions.

Allows persona definition tools to select a manager instance while preserving the caller's existing manager binding and permissions.

Renders a broker TLS mismatch as a concise join refusal with operator guidance instead of a raw connection stack, while keeping the TLS requirement and refusing plaintext connections.

v0.88.0 NOTES ↗ STABLE

Cotal 0.88.0 adds process-runtime presence fields, host-custodied tool-result consent signing, and Manager host/version reporting in verified status.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.88.0 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.88.0
```

Adds `runtime: { id, startedAt, source: 'endpoint-process' }` to presence payloads, letting the OS process holding a mesh connection report its own start time from `performance.timeOrigin`, with a stable opaque `id` shared across status, activity, heartbeats and in-process reconnects.

Adds opt-in host-custodied tool-result consent signing with a closed capability-gated admission door: verifies same-owner password proof and current holder/target lifecycles, then records each signed consent id create-only before release; only the auth mint writer receives consent persistence authority.

Allows harness-less interactive joins on required-events spaces using the signed-in CLI actor and native bearer renewal, without requiring a harness-hosting launch.

Reports the answering Manager's host name and running package version in verified status; exports the `ManagerStatus` type with both advisory fields optional for older status payloads.

└──▷ BREAKING ON UPGRADE

! Managers using a delegated runtime now refuse caller-supplied persona `config` overrides before looking up or reading the requested host path; `resolve-agent` and `spawn` return `failed-precondition` for every path shape, and direct `Manager.startAgent` calls share the refusal. Select a named catalog persona instead.

v0.87.0 NOTES ↗ STABLE

Cotal 0.87.0 adds browser Manager control, a `POST /manager-instances` API, and eliminates consumer-limit errors across CLI commands and history reads.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.87.0 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.87.0
```

└──▷ TRY IT

Check available Manager instances for the signed-in user from the auth service, e.g. to verify registration status before spawning a seat.

```
$ curl -X POST https://<auth-service>/manager-instances \
  -H 'Authorization: Bearer <idp-token>' \
  -H 'Content-Type: application/json'
```

Adds `openBrowserManagerCaller` in `@cotal-ai/core/browser-control` to open a single, strict-CSP-page connection to an explicitly selected Manager instance using a `manager-caller` bearer, with `reopen`, `followGoal`, `reconcileGoal`, and `verifyGoalResultFact` operations, and adds `submitAndFollowGoalWith` (accepting the host's fact verifier), moving goal-follow logic into a transport-node-free module shared between the package root and browser.

Exports `GOAL_TERMINAL_STATES` and `epGoalProgressGrantRow` alongside the existing `submitAndFollowGoal` and `parseGoalResultFact` at the package root.

Adds `directGet` option to `readChannelRegistry` and `readOnce` endpoint option plus `lastKvEntries` to `@cotal-ai/core`, letting `cotal send`, `cotal endpoints`, `cotal personas list --running`, `cotal channels list`, and `cotal spawn` read presence and registry buckets with a single batched Direct Get instead of binding consumers.

The auth service now serves `POST /manager-instances` on both faces, returning `{ v: 1, owner, complete: true, instances }` with placement and verdict-selection fields (`candidate`, `not open`, `not registered`, `not this owner's`, `unknown`); `openAuthAuthorityPlane` gains `listManagerInstances`.

History reads via `channelHistory`, `multiChannelHistory`, and `dmHistory` now use named readers (`hist_<readerId>_<n>`, n in 0..7) that are deleted after each step, capping each connection to eight concurrent history reads per stream, and the observer

profile gains `$JS.API.CONSUMER.DELETE.<CHAT>.hist_<readerId>_<n>` permissions for all eight reader slots, with admin also gaining them on `<DM>`.

+3 more

└──▷ **BREAKING ON UPGRADE**

! A static observer or admin credential minted before this release has no `$JS.API.CONSUMER.DELETE.<CHAT>.hist_<readerId>_<n>` rows; a current client's history read on it fails and instructs the user to remint. Remint static credential files and upgrade the auth service before or alongside hosts that read history on a user-auth mesh.

! The auth callout's `permissionsFor` hook now receives the connection's reader id as a third argument; any `calloutPermissions` function that does not accept it must be updated.

! `materialEnv` in `@cotal-ai/connector-core` now requires a second argument (the launch's artifact array); callers that omit it will break.

v0.86.0 NOTES ↗ STABLE

Cotal 0.86.0 adds issuance auditing for auth callouts, Claude Desktop support, and browser-safe endpoint invocation.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.86.0 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.86.0
```

└──▷ **TRY IT**

Provision a Claude app conversation's mesh identity and write credentials to a local file for use in an agentic session.

```
$ cotal app-identity
```

Remove a conversation's broker footprint and credential file after a session ends without retiring cleanly.

```
$ cotal app-identity retire --dir <path>
```

Serve a remote gateway's tools over stdio so a local MCP server exposes `cotal_connect`, `cotal_wait`, and routed tools without an MCP SDK dependency.

```javascript
import { serveGatewayStdio } from '@cotal-ai/connector-core';
await serveGatewayStdio(host, options);
```

Adds `nativeIssuanceKey` to core and a `recordIssuance` required parameter to `startAuthCallout`, so every auth-callout success writes a create-only `natkey.<userNkey>` row (fields: `space`, `owner`, `actor`, `lifecycle`, `credential`, `view`, `iat` / `exp`) into the space's auth bucket and re-checks connect authority before replying—producing a signed deny on a failed write, duplicate key row, or revocation—plus a new `nativeIssuancePruneMs` config key (default 60 s) that controls how often the authority plane prunes expired `natkey.*` rows from the auth bucket after their `exp` plus clock tolerance.

Adds serveGatewayStdio(host, options) in `@cotal-ai/connector-core` to serve a remote gateway's tools over stdio, letting a local MCP server offer `cotal_connect`, `cotal_wait`, and routed tools without importing an MCP SDK, and adds a `GatewayOptions.wake` hook to signal a connection once when a message arrives, with `cotal_connect` now reporting whether the client will be woken.

Adds hidden `cotal app-identity` plumbing command to open a Claude app conversation's own identity on a selected mesh, provision delivery durables, and write credentials to a 0600 file; `cotal app-identity retire --dir` removes the identity's broker footprint and credential file.

Adds `ConnectorSetupProvider.refresh` in `@cotal-ai/core`; `cotal setup` and `cotal update` now invoke it to bring an installed connector to the current release without installing anything new, and `cotal update` fails when the refresh fails.

Extends `cotal setup` to install the `cotal` Claude Code plugin at user scope (under `~/.cotal/claude-plugin`) so every Claude Code session on the machine gets `cotal_*` tools; previously it installed only for the project `setup` ran in.

+5 more

└─▷ BREAKING ON UPGRADE

- ! `startAuthCallout` now requires a `recordIssuance` argument and refuses to start without one — any caller that omits it will break.
- ! `saveAgentFile` now rejects a `meta` key that names a modeled frontmatter field (`name`, `model`, `owner`, `capabilities`, `channels`, `publish`) — callers that previously passed such keys to overwrite those fields will receive an error instead of silently overwriting them.
- ! `TURN_INPUT_SCHEMA` and `TURN_OUTPUT_SCHEMA` are now deep-frozen — code that mutates either object after import will throw at runtime.

! `cotal setup` now installs the Claude Code plugin at user scope; a repeat `cotal setup` migrates earlier per-project installs to the single user-scope install, which may break projects that relied on project-scoped isolation of the plugin.

v0.85.0 NOTES ↗ STABLE

Cotal 0.85.0 adds a remote MCP gateway surface, a host bearer function for user-mode agents, and ACL-aware `cotal_send` defaulting.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.85.0 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.85.0
```

└──▷ USE IT

Embed the Cotal gateway in your own service so each inbound MCP POST is handled statelessly without bundling a separate MCP SDK.

javascript

```
import { handleGatewayRequest, GATEWAY_INSTRUCTIONS } from '@cotal-ai/connector-core';

app.post('/gateway', async (req, res) => {
  const authResult = await myAuthenticator.verify(req);
  const response = await handleGatewayRequest(req, authResult);
  res.json(response);
});
```

Renew user-mode bearer tokens in-process for agents running inside your service, without shelling out per refresh.

javascript

```
const config: AgentConfig = {
  userAuth: {
    bearer: async () => await tokenStore.getFreshBearer(agentId),
  },
  // other config...
};
```

Adds `handleGatewayRequest` export to `@cotal-ai/connector-core`, serving one stateless HTTP request to a remote gateway endpoint with the package's own MCP server and Streamable HTTP transport; GET and DELETE answer 405, each POST gets a

fresh server and a JSON response, and `GATEWAY_INSTRUCTIONS` is the default server instructions text.

Adds `registerCotalGatewayTools` export to `@cotal-ai/connector-core`, exposing the Cotal tool surface for a remote MCP server that serves many conversations at once via `cotal_connect`; `cotalToolSpecs` accepts an optional set of offered tool names, and `GatewayHost` supplies authentication, credential issuance, revocation and storage.

`AgentConfig.userAuth` now accepts `bearer`, a function returning a fresh bearer, in place of `bearerCmd`, so a service running agents in-process can renew each agent's user-mode credential without executing a command per refresh.

`cotal_send` without a `channel` now defaults to the first channel the agent may both read and post to on ACL-enforced meshes, skipping read-only channels and refusing with a clear message when no postable channel exists.

A launch recipe can now declare `LaunchSpec.confirmReady` — texts confirming the harness reached its input screen — honoured by PTY, tmux, cmux, Orca, and Herdr runtimes; seeing one on two screen reads in a row ends the startup watch with no key.

+2 more

v0.84.0 NOTES ↗ STABLE

Cotal v0.84.0 loads Claude Code's native MCP sources by default and adds a native zsh completion snippet.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.84.0 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.84.0
```

Loads Claude Code's native MCP sources by default alongside Cotal and declared overrides, preserving per-spawn sharing restrictions, carried-session isolation, native policy, and narrow environment forwarding.

Shows separate bash and native zsh snippets in the completion usage banner instead of directing zsh users to the bash script.

└──▷ BREAKING ON UPGRADE

! Claude Code's native MCP sources are now loaded by default — existing spawns that relied on Cotal not loading these sources may pick up additional MCP sources automatically.

v0.82.0 NOTES ↗ STABLE

Cotal 0.82.0 adds hosted Claude Code MCP sessions via `serveClaudeSession` and delegated-child recovery contracts.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.82.0 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.82.0
```

└──▷ USE IT

Host multiple isolated Claude Code MCP sessions from a single process, each reading its environment only from the supplied `env` object.

```javascript
import { serveClaudeSession } from '@cotal-ai/connector-claude-code/mcp';

await serveClaudeSession({
  env: process.env,
  input: incomingStream,
  output: outgoingStream
});
```

Adds serveClaudeSession({ env, input, output }), exported from `@cotal-ai/connector-claude-code/mcp`, to host the Claude Code MCP server over caller-supplied streams so one process can serve several isolated sessions.

Adds `EndpointOptions.environment` to accept an explicit presence label or `null` to omit it without reading `COTAL_ENVIRONMENT`.

Adds delegated-child recovery contracts and closed request validators, binding retained descriptors, scan pages, and claim replies to their lifecycle and manager registration.

Adds `ConnectRefusal` as the thrown type from `connectOrThrow` and `connectUserControlOrThrow` when a user-auth mesh rejects login or exchange (not logged in, auth service down, actor not granted), carrying the provider's sentence.

Launches the per-session MCP server with `--optimize-for-size --max-semi-space-size=1` to reduce its memory footprint.

+1 more

v0.81.1 NOTES ↗ STABLE

Cotal v0.81.1 adds resilient event-plane reconnection, `--permission-mode bypassPermissions` default, and `cotal spawn -f` partial-deploy reporting.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.81.1 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.81.1
```

└──▷ TRY IT

Spawn a managed Claude Code seat with interactive permission prompts re-enabled, overriding the new `bypassPermissions` default.

```
$ cotal spawn --opt permission-mode=default
```

Run a foreground spawn and catch partial failures — exits 1 and reports which launches failed so CI can act on them.

```
$ cotal spawn -f
```

Adds `cotal spawn -f` partial-deploy reporting: exits 1 and reports which launches failed when any launch in a foreground spawn fails.

Adds `--preserve-state` flag enforcement to `cotal down`: refuses a presence row whose card id differs from its key.

Adds `cotal actor revoke` output refinement: prints `verified gone` only on a well-formed eviction reply.

The event plane no longer exits the seat on failure — the connector self-heals by re-establishing the plane immediately after a fault and every second thereafter, recovering pending work once the fault clears.

A torn or unreadable transcript line is now stepped over with its offset and length reported, rather than halting the event reader permanently.

+1 more

v0.79.0 NOTES ↗ STABLE

Cotal v0.79.0 adds `controlReplyFromThrown`, `readProcessRecord`, `CorruptExtensionPackageError`, and `cotal completion` shell-startup fixes.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.79.0 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.79.0
```

└──▷ **TRY IT**

Diagnose a corrupt built-in extension after upgrade and reinstall it without a full reset.

```
$ cotal ext seed --repair
```

Check which pidfile the manager record was read from when debugging a stale or misattributed process record.

```
$ cotal service status --json | jq '.manager.path'
```

Enable shell completion in a dev checkout without triggering the connector seed gate or the source-checkout guard.

```
$ COTAL_SKIP_CONNECTOR_SEED=1 cotal completion bash
```

Adds `controlReplyFromThrown` export from `@cotal-ai/core` so every caller receives a manager call's error `code` and `details` on thrown envelope errors, matching the behaviour of reply-based refusals.

Adds `CorruptExtensionPackageError` (thrown by `installedExtensionVersion`) to surface a damaged extension `package.json` with the package name and file path in `cotal status`, `cotal setup`, and `cotal ext seed --repair`.

Adds `isConsumerNotFound` and exports `isJetStreamMissing` from `@cotal-ai/core` as the single shared predicate for consumer-not-found detection across the stream, KV scan, run wait, delivery `pending`, and auth scanner sites.

Adds `readProcessRecord` to `@cotal-ai/workspace`, returning the record path on every `ProcessRecord`; `service status --json` now reports that path as `manager.path`.

Adds `personaDir` to `@cotal-ai/core` as the single source of truth for the persona catalog directory, consumed by `agentFilePath`, `listPersonaCatalog`, `cotal status`, and `cotal personas`.

+5 more

└──▷ **BREAKING ON UPGRADE**

- ! `isConsumerNotFound` in `@cotal-ai/core` no longer accepts a bare `name` of `ConsumerNotFoundError` or a string `code` — callers relying on those shapes will no longer match consumer-not-found conditions.
- ! A `manager-run-attempt` envelope carrying a `run` field is now refused as an unknown field (previously accepted with the field silently dropped).
- ! The `manager-service` authority parser now refuses a `session` request whose `session` object carries keys other than `id`, `endpoint`, `sessionId`, `epoch`, and `exp`.
- ! `takeCodePoints` is no longer exported from the AG-UI module; callers must replace it with Array.from(s).slice(0, n).join('').

v0.78.0 NOTES ↗ STABLE

Cotal 0.78.0 adds atomic SecretStore.create(), richer `COTAL_TLS` flag spellings, and named `COTAL_CONTROL_SOCKET` / `COTAL_CONTROL_TOKEN` enforcement for Claude Code sessions.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.78.0 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.78.0
```

Adds SecretStore.create(key, value) — an atomic write-if-absent operation that resolves whether it created the entry, preventing concurrent first-callers from minting divergent secrets; `FsSecretStore` implements it with an exclusive create.

Expands `COTAL_TLS` to accept `1`, `true`, `yes`, `on` (case-insensitive) to enable TLS, and `0`, `false`, `no`, `off` to defer to the join link's choice; any other value is refused at startup naming the variable.

Improves `cotal meshes` error reporting: `cotal meshes add --from` now refuses non-JSON discovery documents and reports the URL and content-type that answered instead of the generic `--user-auth-file` message, and `cotal meshes` (and every command that reads the mesh registry) exits 1 with the path and read error when the registry directory exists but cannot be read, instead of silently returning an empty registry.

Adds `COTAL_CONTROL_SOCKET` and `COTAL_CONTROL_TOKEN` to the error message and configuration reference when a Claude Code session is driven without a control endpoint.

Replaces `PrimitiveSpec.opensScope` with `scope` (a `ScopeKind` or `null`), requiring scope rows in `PRIMITIVES` to name a member of `ScopeKind` to compile, and adds a

`dispatches` row to `scopeTraits` so `performScope` reads dispatch behaviour from the table instead of inferring it from `hashesSubject`.

+3 more

└──▷ **BREAKING ON UPGRADE**

! A custom `SecretStore` implementation must now implement create(key, value) as one atomic step; stores that do not will fail to compile or behave incorrectly when concurrent first-callers race to set a key.

WAS THIS USEFUL? 👍 👎

open-multi-agent

Feedback ⓘ v1.22.0 NOTES ↗

Self-hosted TypeScript agent runtime with durable approvals and verifiable run records. Own it, approve it, audit it.

Team.addTask() now honors all task options (retries, priority, memory scope, consensus); MiniMax-M3.1-Flash becomes the default for new configurations.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.22.0 https://github.com/open-multi-agent/open-multi-agent.git
# already have the repo? check out this version:
$ git checkout v1.22.0
```

Team.addTask() now honors every accepted task option — retry settings, memory scope, dependency payload, role, priority, metadata, capability requirements, and consensus verification — affecting queue ordering, retries, capability gating, memory scope, and consensus verification.

NEW INTEGRATION New MiniMax configurations default to MiniMax-M3.1-Flash; saved configurations, explicit MiniMax-M3, legacy IDs, and custom model IDs continue to be forwarded unchanged.

The `oma` provider reference and default-model hint for MiniMax now name MiniMax-M3.1-Flash, so newly generated provider templates use that default.

The MiniMax adapter now reports tool-use-only native reasoning echo support for follow-up requests.

└──▷ BREAKING ON UPGRADE

! Team.addTask() now validates and acts on all accepted task options (retry settings, memory scope, dependency payload, role, priority, metadata, capability requirements, consensus verification); options previously silently discarded now affect queue ordering, retries, capability gating, and can add model calls and cost. Malformed metadata that was previously ignored now causes addTask() to throw before the task is enqueued — callers must ensure metadata satisfies `TaskMetadata`, including homogeneous arrays.

! MiniMax-M3.1-Flash cannot disable thinking: a call with `thinking: { enabled: false }` now throws before the request rather than forwarding silently. Use MiniMax-M3 when disabled thinking is required, or pass an explicit `extraBody.thinking` override.

WAS THIS USEFUL? 👍 👎

Stacklok Mecatl

Feedback ⓘ v0.0.50 NOTES ↗ RANK

Mecatl is an open-source agent harness for running the same provider-agnostic agent loop locally, remotely, or on Kubernetes, with durable state, permissions, and gRPC/HTTP-SSE APIs.

> Mecatl v0.0.50 adds an identity broker subsystem for agent credential issuance in mecak8s, a scheduler toggle for the mecak8s Helm chart, and a unified permission-mode vocabulary with new admission gates.

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

01 **Scheduler toggle for mecak8s Helm chart** NEW 70

Adds a `scheduler.enabled` chart value to the `mecak8s` Helm chart to toggle the scheduler component, e.g. setting `scheduler: enabled: true` in the Helm values.

Enable the scheduler in a mecak8s Helm deployment so agent sessions can be scheduled.

yaml

```yaml
scheduler:
  enabled: true
```

— Exact config key and runnable Helm value given v0.0.50

02 **Identity broker subsystem for mecak8s** NEW 60

Introduces an identity broker subsystem for `mecak8s` that registers agents at session start, issues their credentials, and opens/closes broker sessions tied to Mecatl sessions, running behind an authentication boundary.

— Mechanism described but no API, flag, or config name given v0.0.50

THINNER COVERAGE BELOW

03 **Permission-mode vocabulary and admission gates** NEW 20

Adds a unified permission-mode vocabulary and two admission gates to the permissions subsystem.

— No names for the modes, gates, or config surfaces v0.0.50

WAS THIS USEFUL? 👍 👎

Microsoft Agent Framework

Feedback ⓘ dotnet-1.26.0

NOTES ↗ RANK

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

> This window's biggest changes are breaking: an MCP 2.x compatibility migration spanning seven packages, a renamed Docker shell config key, and mandatory AG-UI authorization scopes, alongside new .NET session deletion, gRPC A2A transport, Anthropic multi-tool-call support, and Python 3.15 support for DuckDB, Hyperlight, and the AG-UI A2UI extra.

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

01 MCP 2.x compatibility and client lifecycle management

BREAKING

80

Adds an MCP 2.x compatibility foundation across `agent-framework`, `agent-framework-ag-ui`, `agent-framework-core`, `agent-framework-declarative`, `agent-framework-foundry-hosting`, `agent-framework-hosting-mcp`, and `agent-framework-lab`, migrating MCP transports, errors, schemas, and lifecycle handling; this is a breaking change and existing MCP integrations must be updated. Also defines caller-owned and framework-owned MCP client lifecycles in `agent-framework-core` and `agent-framework-devui`.

— Names every affected package and lifecycle split; no migration steps given.

python-1.23.0 dotnet-1.26.0

02 Session deletion via `DeleteSessionAsync`

NEW

80

The .NET `AgentSessionStore` gains `DeleteSessionAsync`, letting callers programmatically delete a session by ID and free server-side state, e.g. `await sessionStore.DeleteSessionAsync(sessionId, cancellationToken);`.

Delete a session by ID at the end of a conversation to free server-side state, using the new `DeleteSessionAsync` method on `AgentSessionStore`.

csharp

```
await sessionStore.DeleteSessionAsync(sessionId,
cancellationToken);
```

— Runnable code example with exact method signature. dotnet-1.26.0

03 Multi-tool calls in Anthropic client via `allow_multiple_tool_calls`

NEW

80

Python `AnthropicChatClient` adds an `allow_multiple_tool_calls` flag that lets an agent call multiple tools per turn when no explicit `tool_choice` is set, e.g. AnthropicChatClient(model="claude-opus-4-5", allow_multiple_tool_calls=True).

Allow an Anthropic-backed agent to call multiple tools per turn when no explicit tool_choice is set, using the `allow_multiple_tool_calls` flag.

```python
agent = Agent(
    client=AnthropicChatClient(
        model="claude-opus-4-5",
        allow_multiple_tool_calls=True,
    ),
    name="MultiToolAgent",
    instructions="You are a helpful assistant.",
)
```

— Runnable code example with exact flag name. `dotnet-1.26.0`

---

04 **Docker shell config key renamed for safety** BREAKING

73

The Docker shell config key `extra_run_args` in `agent-framework-tools` is renamed to `unsafe_extra_run_args`; existing configurations using `extra_run_args` will break. Docker params usage was also clarified in both .NET and Python (PR #9453).

— Exact old and new config key names given, clear migration path. `python-1.23.0` `dotnet-1.26.0`

---

05 **Python 3.15 support for DuckDB, Hyperlight, and AG-UI A2UI extra**

NEW

70

Adds Python 3.15 support across `agent-framework-duckdb` (requiring DuckDB 2.0), `agent-framework-hyperlight` (requiring Hyperlight 0.8), and the AG-UI A2UI extra in `agent-framework-ag-ui` (requiring A2UI toolkit 0.0.5).

— Names exact packages and minimum dependency versions. `python-1.23.0`

---

06 **Mandatory AG-UI authorization scope** BREAKING

68

AG-UI endpoint registration in `agent-framework-ag-ui` and `agent-framework-core` now requires an explicit AG-UI authorization scope or a single-user declaration; endpoints registered without one will fail, breaking existing setups that omit the scope.

— Names the packages and exact failure condition, no code sample. `python-1.23.0`

dotnet-1.26.0

THINNER COVERAGE BELOW

07 **Opt-in synthetic trailing user turn** NEW 55

`agent-framework-core` and `agent-framework-orchestrations` add an opt-in synthetic trailing user turn for providers that return empty text after assistant-ending prompts.

— Names the packages and trigger condition, no flag name given. python-1.23.0

08 **A2A request metadata forwarding in Python** IMPROVED 52

`agent-framework-a2a`'s `A2AExecutor` forwards run kwargs from A2A request metadata, with an allowlist controlling which function-invocation and client arguments are forwarded.

— Names package and executor but not the allowlist contents. python-1.23.0 dotnet-1.26.0

09 **gRPC transport for .NET A2A protocol** NEW 40

The .NET A2A sample adds gRPC protocol selection, enabling gRPC as an alternative transport alongside HTTP for agent-to-agent communication.

— Sample-level addition, no API surface named. dotnet-1.26.0

10 **Fan-out deduplication in .NET workflows** IMPROVED 35

.NET workflows now deduplicate fan-out targets so each target runs exactly once, preventing duplicate executions.

— Explains the behaviour change but no API or config named. dotnet-1.26.0

11 **AG-UI request-time agent selection and hosted-agent lifetimes** IMPROVED 33

AG-UI hosting in .NET adds request-time agent selection, allowing the agent to be chosen per request, alongside honoring hosted-agent lifetimes.

— Describes the capability but no config or API named. dotnet-1.26.0

12 **Warning log for skipped declarative actions** IMPROVED 28

The .NET declarative engine now logs a warning when an unsupported declarative action is skipped instead of failing silently.

— Brief behaviour note, no specifics on which actions. dotnet-1.26.0

13 **File-search timeout configuration in .NET client** NEW 23

The .NET client adds a configurable timeout for file-search operations.

— No flag name, default, or usage example given. `dotnet-1.26.0`

╰──▷ BREAKING ON UPGRADE

- ! Docker params usage clarified and a param renamed in both .NET and Python (PR #9453, tagged [BREAKING]).
- ! Python now requires explicit AG-UI authorization scope; existing setups that omit the scope will break (PR #9413, tagged [BREAKING]).
- ! Python MCP 2.x compatibility foundation introduces breaking changes to the MCP client API (PR #9236, tagged [BREAKING]).
- ! MCP 2.x compatibility migration in `agent-framework`, `agent-framework-ag-ui`, `agent-framework-core`, `agent-framework-declarative`, `agent-framework-foundry-hosting`, `agent-framework-hosting-mcp`, and `agent-framework-lab` — MCP transports, errors, schemas, and lifecycle handling have changed; existing MCP integrations must be updated.
- ! AG-UI endpoint registration in `agent-framework-ag-ui` and `agent-framework-core` now requires an explicit AG-UI authorization scope or a single-user declaration; endpoints registered without one will fail.
- ! Docker shell config key `extra_run_args` in `agent-framework-tools` is renamed to `unsafe_extra_run_args`; existing configurations using `extra_run_args` will break.

WAS THIS USEFUL? 👍 👎

trigger.dev

Feedback ⓘ v4.8.1 NOTES ↗ RANK

Trigger.dev – build and deploy durable AI agents and workflows

> Trigger.dev's biggest change this window is a named, pausable concurrency-limits API with per-key temporary overrides, shipped alongside query-watching alerts with MCP management, a code-driven custom dashboards API, a new CLI command to trigger tasks from the terminal, and a wide set of CLI onboarding, chat-agent, and self-hosted hardening improvements.

└──▷ **WHAT SHIPPED · 25 FEATURES** most completely described first ? what's the number?

01 **Named concurrency limits API** NEW 98

concurrencyLimit() creates named, shared concurrency limits configured via the `concurrency` option on task() (an inline shape or up to two named limits, where `perKey` caps per-key pools and `total` caps globally); concurrencyLimits.pause(name) and concurrencyLimits.resume(name) stop and restart dequeuing of runs holding a named limit without changing its bounds, and concurrencyLimits.list(), retrieve(name), override(name, { perKey, total }), and reset(name) manage limits at runtime. concurrencyLimits.override(name, { concurrencyKey, perKey, ttl }) can also scope a temporary override to a single `concurrencyKey` (7-day default, 30-day maximum), with reset(name, { concurrencyKey }) to clear it and retrieve(name, { concurrencyKey }) to inspect the bound and expiry. queues.list() and queues.retrieve() now return a `version` field (`V1` or `V2`) distinguishing legacy queue-level concurrency from the new task-level model.

Cap a task at 25 concurrent OpenAI calls globally and 1 per user, sharing the named limit across multiple tasks.

typescript

```typescript
import { concurrencyLimit, task } from "@trigger.dev/sdk";

export const openaiLimit = concurrencyLimit({ name: "openai",
total: 25 });

export const generateSummary = task({
  id: "generate-summary",
  concurrency: [{ perKey: 1, total: 5 }, openaiLimit],
  run: async (payload: { userId: string; text: string }) => {
    // ...
  },
});
```

Pause all runs competing for an OpenAI concurrency limit during an outage, then resume once service is restored.

```typescript
import { concurrencyLimits } from "@trigger.dev/sdk";

// During incident
await concurrencyLimits.pause("openai");

// After recovery
await concurrencyLimits.resume("openai");
```

Temporarily lower the `perKey` concurrency for one tenant that is hammering a shared limit, then inspect the override to confirm expiry.

```typescript
import { concurrencyLimits } from "@trigger.dev/sdk";

// Cap this tenant to 2 concurrent runs under the shared limit
for 1 day
await concurrencyLimits.override("shared-limit", {
  concurrencyKey: "tenant_abc",
  perKey: 2,
  ttl: 86400,
});

// Inspect the active override
const info = await concurrencyLimits.retrieve("shared-limit", {
  concurrencyKey: "tenant_abc",
});
console.log(info);
```

— Full mechanism, named methods, and two runnable examples v4.7.4 v4.7.0

---

02 **CLI command to trigger tasks from the terminal** NEW

88

`trigger.dev tasks trigger <task id>` CLI subcommand triggers a task from the terminal with `--env`, `--wait`, and `--payload` flags, optionally waiting for its output and exiting with code 1 on failure.

Trigger a task in production from CI and block until it succeeds or fails, surfacing the output or error in the terminal.

```
$ npx trigger.dev@latest tasks trigger hello-world --payload
'{"name":"Ada"}' --env prod --wait
```

— Runnable command with flags and a worked example v4.7.5

03 **Code-driven custom dashboards API** NEW 88

`query.dashboards` SDK API lists, creates, updates, and deletes custom metric dashboards and their widgets from code, validating widget queries on save; MCP server tools `create_dashboard` and `set_dashboard_widgets` let AI assistants build dashboards programmatically.

Create a custom metrics dashboard with a run-count widget entirely from code, no UI required.

typescript

```typescript
import { query } from "@trigger.dev/sdk";

await query.dashboards.create({
  title: "Email tasks",
  widgets: [
    {
      title: "Total runs",
      query: "SELECT count() AS total FROM runs",
      display: { type: "bignumber", column: "total",
aggregation: "sum" },
    },
  ],
});
```

— Named SDK API, MCP tools, and a runnable example v4.7.5

04 **Span redaction and chat telemetry controls** NEW INTEGRATION NEW 86

`telemetry.spanProcessors` in `trigger.config.ts` runs custom span processors before every export, enabling redaction or dropping of span content before it reaches the run trace; `chat.agent`'s new `telemetry` option adds `recordUserMessage` and `recordClientData` fields to stop recording sensitive data on agent spans.

Redact sensitive span content before it leaves the worker, and suppress recording of the user message on a chat agent.

```typescript
export default defineConfig({
  telemetry: { spanProcessors: [new MyRedactingSpanProcessor()]
},
});

chat.agent({
  id: "my-chat",
  telemetry: { recordUserMessage: false, recordClientData: false
},
  run: async ({ messages, signal, streamText }) =>
    streamText({ model, messages, abortSignal: signal }),
});
```

— Names config key, fields, and includes a worked example `v4.7.5`

05 **Run replay with override options** NEW 86

runs.replay() accepts new `payload`, `metadata`, `tags`, `queue`, `machine`, `version`, or `delay`, falling back to the original run for omitted fields and preserving `Date` and `BigInt` types; a `replay_run` MCP tool exposes the same options to AI assistants, hidden automatically in `--readonly` mode.

Replay a failed run with a corrected payload and a 10-minute delay so downstream rate limits clear before retry.

```typescript
import { runs } from "@trigger.dev/sdk";

await runs.replay("run_1234", {
  payload: { userId: "user_456" },
  tags: ["replayed"],
  delay: "10m",
});
```

— Names SDK method, MCP tool, and a worked example `v4.7.4`

06 **Deploy and promote environment targeting rules** BREAKING 81

`trigger deploy` and `trigger promote` gain `--prod` as shorthand for `--env prod`, now default to the environment matching the `TRIGGER_ACCESS_TOKEN` key instead of assuming prod when `--env` is omitted, and error when a passed `--env` mismatches that key; the MCP `deploy` tool no longer defaults to prod and requires an explicit environment.

— Names exact flags and the breaking default change v4.7.5

07 **Self-hosted sign-in security settings** NEW 79

Self-hosted instances can add a Cloudflare Turnstile bot check to the magic-link sign-in form by setting `TURNSTILE_SITE_KEY` and `TURNSTILE_SECRET_KEY` (off unless both are set), disable the default same-browser restriction on magic links with `MAGIC_LINK_SAME_BROWSER_REQUIRED=false`, and block sign-in and sign-up from specific email domains with the `BLOCKED_EMAIL_DOMAINS` setting.

Enable Cloudflare Turnstile bot protection on the magic-link sign-in page for a self-hosted instance.

```ini
TURNSTILE_SITE_KEY=0x4AAAAAAAxxxxxxxxxxxxxxx
TURNSTILE_SECRET_KEY=0x4AAAAAAAyyyyyyyyyyyyyyyyy
MAGIC_LINK_SAME_BROWSER_REQUIRED=false
```

Prevent sign-ups and sign-ins from specific email domains on a self-hosted instance.

```ini
BLOCKED_EMAIL_DOMAINS=example.com,spam.org
```

— Names three env vars with a working config example Named env var with a working example value

v4.7.4 v4.7.3

08 **CLI onboarding, authentication and project setup** IMPROVED 78

`trigger.dev orgs list` lists organizations with slugs and IDs, usable with `trigger.dev projects create --org`; `trigger.dev projects create` now auto-activates the Free plan for organizations without one, matching `trigger.dev init` behavior; `init` can create the first organization, activate its Free plan, and scaffold the first project; `login` can prefill email and resume authorization; `trigger promote` can authenticate via environment API keys supplied through `TRIGGER_ACCESS_TOKEN`; interactive `init <path>` writes a custom task directory to `trigger.config.ts` relative to the project directory; and the CLI/MCP authorization page now shows the account and organizations being granted access.

Look up your org slug before creating a new project, especially for orgs with no existing projects.

```
$ trigger.dev orgs list
```

— Enumerates many named commands and settings v4.7.3 v4.7.1 v4.7.0

09 **Stricter permissions for session creation endpoint** BREAKING

74

Creating or resuming a session with `POST /api/v1/sessions` now requires both session-write and task-trigger permissions; scoped keys and public tokens must grant `write:sessions` (or access to the target session) and permission to trigger its task.

— Names exact endpoint and required scopes v4.7.3

10 **Query-watching alerts with MCP management** NEW

73

Alerts can now watch any supported query result and fire on threshold crossings, missing matching rows, or new matching rows, creatable from the Query page, Dashboard Agent, or API, with alert values charted on the alert's page. MCP clients gain `list_alert_capabilities`, `list_alerts`, `create_alert`, and `resolve_alert` tools, with health thresholds discoverable via `list_alert_capabilities` and settable via `create_alert`.

— Names triggers, surfaces, and four MCP tools, no example v4.8.1 v4.7.6

11 **Self-hosted API rate-limit metrics** NEW

72

Self-hosted instances now record API rate limit usage per environment in the `metrics` table as `api.rate_limit.allowed`, `api.rate_limit.denied`, `api.rate_limit.remaining_min`, `api.rate_limit.limit.per_second`, and `api.rate_limit.limit.burst`, queryable on the Query page and dashboards.

— Names five exact metric fields and location v4.7.0

12 **Self-hosted deploy base image overrides** NEW INTEGRATION NEW

72

Self-hosted instances can enforce custom deploy base images per runtime via new `DEPLOY_BASE_IMAGES` and `DEPLOY_BUILD_BASE_IMAGES` webapp settings; the CLI builds on the images the instance specifies and rejects older CLIs with an upgrade message.

— Names both settings and the CLI enforcement behavior v4.7.4

13 **New session-scoping parameters for chat start actions** NEW

68

Adds `concurrencyKey` and `triggerConfig.concurrency` to chat.createStartSessionAction() and the `AgentChat` client to scope concurrency per session (for example per chat ID or tenant ID), and a `tags` parameter to `chat.createStartSessionAction`, `chat.headStart`, and `chat.startHeadStart` to let chat sessions be filtered by tag on the Sessions page.

Scope a chat agent's concurrency to individual users so one busy session cannot block others.

```typescript
import { chat } from "@trigger.dev/sdk";

const start = chat.createStartSessionAction("support-chat", {
  triggerConfig: { concurrencyKey: user.id },
});
```

Tag a chat session at creation time so it can be filtered by user or org on the Sessions page.

```typescript
await startChatSession({
  chatId,
  clientData,
  tags: [`org:${org.slug}`, `user:${user.id}`],
});
```

— Names fields and client, includes a worked example Names parameter and three functions, with example

v4.7.0 v4.7.3

14 **Redeploy from the deployments list** IMPROVED 61

Deployments made with `--local-bundle` can be redeployed from the deployments list, rebuilding the same bundle with its build-time environment variables, and deployments started from the GitHub integration or a native CLI deploy can likewise be redeployed, rebuilding the same commit or upload as a new deployment.

— Covers three deploy sources via a UI path only v4.7.4 v4.7.5

15 **Webhooks dashboard visibility and rate limiting** NEW 61

Webhooks now appear in the dashboard side menu with a Beta badge, with organizations lacking access able to request beta access from the Webhooks page; each webhook endpoint's request rate limit now comes from the organization's plan and can be raised per organization, shown on the limits page, and each environment's webhook endpoints share a rate limit that counts larger request bodies as more than one request, also configurable per organization's plan.

— Covers visibility and two rate-limit mechanisms v4.7.4

THINNER COVERAGE BELOW

16 **submit_feedback MCP tool** NEW 57

A `submit_feedback` MCP tool lets coding agents report tool errors, docs mismatches, or missing capabilities; disable it with `--skip-telemetry` or `TRIGGER_TELEMETRY_DISABLED`.

– Names tool and two disable mechanisms, no example v4.7.0

---

17 **Preview deploy branch detection on detached-HEAD CI** IMPROVED 56

Preview deploys now detect the current branch on CI providers that check out a detached HEAD — GitLab, Bitbucket, CircleCI, Buildkite, Travis CI, Jenkins, and Azure Pipelines — instead of creating a branch called `HEAD`.

– Lists all seven CI providers covered, no example v4.7.5

---

18 **API key access for waitpoint tokens** NEW 56

API keys with waitpoint access can now create, list, and retrieve waitpoint tokens; the new API key dialog's access summary shows Errors, Webhooks, and Waitpoints access.

– Names access categories but no example call v4.7.4

---

19 **TLS and certificate handling fixes in deploy images**

UPDATED SUPPORTED PLATFORM IMPROVED 52

`NODE_EXTRA_CA_CERTS` set as an environment variable now reaches the task process at startup instead of being overwritten by the image's empty default, and Python tasks and `pip` in deploy images now verify TLS against the image's trust store instead of a pinned certifi bundle.

– Precise before/after, but only an upgrade action v4.7.6

---

20 **Clear error for unsupported deployment runtimes** IMPROVED 52

Deployments with an unsupported `runtime` now return a `400` response naming the supported runtimes instead of a generic `500`.

– Names status codes but no example response v4.7.3

---

21 **trigger dev coding-agent mode** IMPROVED 50

`trigger dev` now skips install prompts, lists registered tasks, and prints each run's dashboard URL and output when run by a coding agent (Claude Code, Cursor, Codex).

– Describes behavior but no exact invocation shown v4.7.5

22 **Queue archiving on the Concurrency page** NEW 48

Self-hosted instances can archive unused queues from the Concurrency page so they are hidden from lists and excluded from allocated concurrency counts.

— Thin description, only a UI path given v4.7.0

23 **Chat transport reliability: callback and stall diagnostic** NEW 46

TriggerChatTransport.sendAction() gains an `onSettled` callback to confirm an action's input was processed independently of whether the response stream closes, and chat streams now emit a `'Stream stalled: no records received'` diagnostic after five retries of a connected stream that sends no records, covering silent tool calls and browser wakeups.

— Names callback and diagnostic message, no example v4.7.0

24 **MCP tool discovery fix for date fields** IMPROVED 39

Zod 4 z.date() fields are now converted to date-time strings in JSON Schema so MCP tool discovery no longer fails when a tool input schema contains a date.

— Clear fix but no reader action required v4.7.0

25 **Progressive run trace loading** IMPROVED 29

Run traces now appear almost immediately and load progressively, with support for much larger traces than before.

— Vague improvement with no numbers given v4.7.3

└──▷ BREAKING ON UPGRADE

! Passing `--env` to `trigger deploy` or `trigger promote` that does not match the environment of the `TRIGGER_ACCESS_TOKEN` key is now an error; previously the deploy silently targeted the key's environment.

! The MCP `deploy` tool no longer defaults to prod; the environment must now be explicitly chosen.

! Creating or resuming a session with `POST /api/v1/sessions` now requires both session-write and task-trigger permissions. Scoped keys and public tokens must grant `write:sessions` (or access to the target session) and permission to trigger its task.

v4.8.0 NOTES ↗ STABLE

Adds hosted webhooks with waiter primitives, chat channel routing, per-turn telemetry spans, resilient realtime streams, and a `TRIGGER_DASHBOARD_URL` env var for self-linking runs.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.8.0
https://github.com/triggerdotdev/trigger.dev.git
# already have the repo? check out this version:
$ git checkout v4.8.0
```

Adds hosted webhook endpoints via webhooks.endpoint.define({ id, source }) — Trigger.dev hosts the URL, verifies every request, and routes events to subscribers; subscribe tasks with webhook({ endpoint, filter, onEvent }) and pause a run until a matching event arrives with webhooks.waiter.create() and webhooks.waiter.wait().

Adds `additionalSignatureHeaders` to HMAC webhook verifier configs to accept alternative signature headers during provider secret rotations.

Adds `generate_webhook_secret` and `set_webhook_secret` MCP tools with an optional `previousSecretTtlSeconds` parameter (up to 7 days) to keep the old signing secret valid while a provider switches to the new one.

Adds chat.event({ endpoint, key }) to route verified webhooks into agent sessions and chat.channels.custom({ endpoint }) to connect external chat surfaces to a shared webhook endpoint; channels support reply delivery, session binding, conversation seeding, lifecycle reactions, `errorMessage`, stream(), lifecycle(), progress(), onControl(), download(), received(), `pendingMessages`, and approval/tool-control resume flows.

Adds chat.createWatchToken() to watch an agent session from another surface.

+7 more

v4.7.2 NOTES ↗ STABLE

`chat.agent` now preserves tool-call order for Claude models and recovers long responses without blocking the worker.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.7.2
https://github.com/triggerdotdev/trigger.dev.git
# already have the repo? check out this version:
$ git checkout v4.7.2
```

`chat.agent` now sends parallel tool results to the model in the order the tools were called, preventing Claude Sonnet 5.5, Claude Opus 5.5, and Claude Fable 5.1 from losing a turn's reasoning or failing requests on accounts that enforce conversation-order checks.

> Recovering or capturing a long `chat.agent` response on a continuation run now rebuilds from streamed chunks in linear time while yielding to the event loop, keeping heartbeats alive so runs are not killed mid-replay.

WAS THIS USEFUL? 👍 👎

LangChain LangGraph

Feedback ⓘ `cli==0.4.38` NOTES ↗ RANK

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

> LangGraph's CLI and SDK both gained priority controls, letting teams configure scheduling priority for deployments and threads.

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

01 **Priority configuration for deployments and threads** NEW

55

LangGraph CLI 0.4.38 adds a `priority` section to `langgraph.json` to configure deployment priority settings, while LangGraph SDK 0.4.9 adds a `Priority` class to the Python SDK for configuring thread priority settings.

— Names exact config key and SDK class but not mechanism or priority values `cli==0.4.38` `sdk==0.4.9`

WAS THIS USEFUL? 👍 👎

Agno (formerly Phidata)

Feedback ⓘ v3.1.10 NOTES ↗

RANK

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

> Agno's biggest release in this window is CompanyOS, a new layer for autonomous, human-gated multi-agent work orchestration, alongside smaller AgentOS improvements to tracing, external agent support, and knowledge API validation.

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

01 CompanyOS: autonomous agent company orchestration NEW

100

Enabled by passing company=CompanyConfig() to `AgentOS`, adding Company routes, permission scopes, and a Control Plane Company page. Introduces Product (a named area of agent responsibility with `sources`, `routines`, `purpose`, and outcome goals) as the top-level unit of work organization, Routine (with `cron_expr`, `timezone`, `executor_type`, `executor_id`, `instructions`, and `auto_accept` fields) to schedule recurring discovery and proposal of Projects and Tasks, and Source (with `type`, e.g. `'markdown'`, and `reference` fields) so Routines can read local Markdown files, local Git repositories, and websites. Executing accepted Tasks requires queue=QueueConfig(durable=True, db=db), with SQLite supported for local development and PostgreSQL for production. Two conversational agents are introduced: Company Manager, for creating Products and Routines, reviewing proposed work, and checking approval queues, and OS Manager, for building and registering the agents that execute approved Tasks. Finished Tasks move to an In review state as a human-approval gate; Routines with `auto_accept` can self-accept Tasks only within configured executor, scope, count, and expiry limits. Any local agent, team, or workflow registered in AgentOS — including sandboxed Claude and Codex agents — can serve as a Task executor, selected at accept time or in a Routine definition.

Bootstrap CompanyOS with a Customer Feedback Product and a daily Routine that proposes Projects and Tasks each morning for human review.

python

```python
from pathlib import Path
from agno.company import Product, Routine, Source
from agno.job_queue import QueueConfig
from agno.models.openai import OpenAIResponses
from agno.os import AgentOS, CompanyConfig, OSManager
from agno.db.postgres import PostgresDb

db = PostgresDb()
model = OpenAIResponses(id="gpt-6.1-sol")

feedback = Product(
    name="Customer Feedback",
    purpose="Turn customer feedback into useful product
improvements.",
    sources=[
        Source(
            type="markdown",
            reference=str(Path("feedback").resolve()),
        ),
    ],
    routines=[
        Routine(
            name="review-feedback",
            executor_type="agent",
            executor_id="company-manager-internal",
            cron_expr="0 9 * * *",
            timezone="UTC",
            instructions=(
                "Read customer feedback and check existing
Projects and Tasks. "
                "Identify recurring bugs, gaps, and feature
requests. "
                "Propose focused Projects and Tasks with
supporting evidence. "
                "Reuse existing work and leave proposals for
human approval. "
                "If nothing needs attention, create no work."
            ),
            auto_accept=False,
        ),
    ],
)

agent_os = AgentOS(
    db=db,
    manager=OSManager(model=model),
```

```
    company=CompanyConfig(model=model, products=[feedback]),
    queue=QueueConfig(durable=True, db=db),
    scheduler=True,
)
```

— Full mechanism, named config keys and fields, runnable bootstrap example

launch-20261023-c3adecf4

02 **Tracing span attribute limit raised to 1,024** IMPROVED

85

AgentOS(tracing=True) and setup_tracing() now retain up to 1,024 span attributes, preserving full model input and message history on runs with many tool calls; the `OTEL_SPAN_ATTRIBUTE_COUNT_LIMIT` and `OTEL_ATTRIBUTE_COUNT_LIMIT` environment variables still take precedence when set.

— Names exact config and override env vars, no example given v3.1.10

03 **External agent discovery and auth for Claude/Codex** NEW

78

OS Manager can list and run external agents such as `ClaudeAgent` and `CodexAgent` registered via AgentOS(agents=[...]), addressable by name or ID. `CodexAgent` now authenticates automatically using `OPENAI_API_KEY` while preserving any existing ChatGPT or other login session.

Register external agents with AgentOS so OS Manager can discover and invoke them by name or ID at runtime.

python

```python
from agno.agent_os import AgentOS
from agno.agent.codex import CodexAgent
from agno.agent.claude import ClaudeAgent

os = AgentOS(
    agents=[ClaudeAgent(), CodexAgent()],
    tracing=True,
)
os.serve()
```

— Named classes and env var with a runnable registration example v3.1.10

04 **Validation error on knowledge content upload** IMPROVED

70

`POST /knowledge/content` now returns a clear `400` error when an upload omits a file, `text_content`, or URL, or when `url` is not a single URL.

— Named endpoint and fields but no usage example v3.1.10

WAS THIS USEFUL? 👍 👎

sortie

Feedback ⓘ v1.30.0 NOTES ↗ RANK

Turn tracker tickets into autonomous agent sessions

> Sortie's six releases this window centered on cost control and observability — spend ceilings at multiple scopes, a live price table, and a `GET /api/v1/prices` / `GET /api/v1/stats` pair of endpoints feeding a new client-side dashboard — alongside label-driven multi-stage dispatch (with a breaking change to label-gated dispatch) and a deprecated `kiro` agent kind replaced by `agent-client-protocol` with Gemini CLI token tracking.

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

01 **`agent-client-protocol` agent kind replacing deprecated `kiro`**

DEPRECATED 95

Adds `agent-client-protocol` as the replacement kind for the deprecated `kiro` kind in `WORKFLOW.md` — set `agent.kind: agent-client-protocol` with `command: kiro-cli acp -a` or `command: gemini --acp --skip-trust --approval-mode yolo`; adds a usage-source reader for Gemini CLI so token usage is tracked and reported when running as an `agent-client-protocol` agent.

Run Gemini CLI as an unattended agent-client-protocol agent with token usage tracking now available in v1.25.0.

yaml

```yaml
agent:
  kind: agent-client-protocol
  command: gemini --acp --skip-trust --approval-mode yolo
  max_turns: 15
  max_concurrent_agents: 4
  turn_timeout_ms: 1800000
```

Migrate a Kiro workflow from the deprecated `kiro` kind to `agent-client-protocol` to stay on a supported route.

yaml

```yaml
agent:
  kind: agent-client-protocol
  command: kiro-cli acp -a
  max_turns: 15
  max_concurrent_agents: 4
  turn_timeout_ms: 1800000
```

— Exact config keys and migration commands with examples v1.25.0

02 **Model pricing and cost/token usage visibility across API, Prometheus, history, and dashboard**

NEW 90

Adds an embedded, runtime-refreshable model price table with validation, lookup, and a one-command refresh, exposed via a new `GET /api/v1/prices` endpoint and a dashboard Prices screen that shows the rates in force; the Prices screen also charts spend over time with cost per succeeded run and cost per model, the run-stats API serves per-model cost parts, partial-run counts, and spend split by UTC day and month, Prometheus counters for usage, runtime, and exit are labeled by model, the dashboard shows budget tokens excluding cache reads, and run history records estimated cost and token usage per model for each run.

Check the model rates currently loaded by a running Sortie instance — useful before setting spend ceilings.

```
$ curl http://localhost:8642/api/v1/prices
```

— Named endpoint plus a runnable curl example Many named surfaces but no exact endpoint or UI path

v1.29.0

03 **`GET /api/v1/stats` run-statistics endpoint** NEW 90

Adds `GET /api/v1/stats`, serving run statistics broken down by what started each run, with per-stage-chain funnel reporting and counts of failed, stopped, and cancelled runs.

Pull run statistics broken down by trigger source and stage-chain funnel to feed a cost or throughput dashboard.

```
$ curl http://localhost:8642/api/v1/stats
```

— Named endpoint with a runnable curl example v1.28.0

04 **One-shot CI runs via `--once`/`--once-result`** NEW 80

`--once` exits the Sortie process once all dispatched work finishes, enabling one-shot CI-style runs, and `--once-result` writes each issue's outcome as JSON for downstream pipeline consumption of agent results.

— Names exact flags and their behaviour, no command example v1.30.0

05 **Expanded state API and component health reporting** NEW

73

Expands the state API to expose retry triggers and parked issues, pending PR reactions and a `check-now` trigger for pending pull-request reactions, the effective workflow shape, run history grouped by issue with counts of unpriced runs, and health, version, and start time; records health of database, poll, and provider components individually; and lets an operator trigger a scheduled retry immediately on request.

— Many named fields but no concrete path or payload shown v1.28.0

---

06 **Skill-call tracking across run stats, session view, and dashboard**

NEW

70

Counts skill calls per skill in the run statistics report, adds a Skills panel to the session view, shows Approval status wherever an issue waits on approval, shows the tracked project on the web dashboard, surfaces skill calls on the History and Analytics views, and serves the tracked project plus each tool call's line and the run's skill calls via the state API.

— Several named surfaces, no exact endpoint path given v1.30.0

---

07 **Client-side dashboard with session view and attention list** NEW

70

Replaces the server-rendered dashboard with a client-side app served under `/ui/`, adds a per-issue session view with live agent events and the agent's last message, and adds an attention list for items that need a human operator, with operator-acknowledgement support.

— Gives a UI path but no deeper config detail v1.28.0

---

08 **Multi-stage issue progression via stage labels** NEW

70

Automatically advances an issue to the next stage label after a successful agent run, enabling multi-stage pipelines; selects the dispatch rule to apply based on the stage label currently on an issue; records stage transitions in run history and events for full pipeline audit trails; and lets tracker adapters remove a label and confirm label writes, giving dispatch rules precise label lifecycle control.

— Mechanism explained but no config example shown v1.27.0

---

09 **Spend ceilings at weekly, daily, monthly, and per-issue scope**

NEW

65

Adds a weekly instance-wide spend ceiling alongside per-issue money ceilings and instance-wide daily and monthly spend ceilings to bound agent cost at multiple scopes, with instance spend ceilings and per-issue money ceilings shown on the dashboard Overview.

— Scopes are named but no config key or exact limit given v1.30.0 v1.29.0

10 **Richer run-history data model with dispatch context** NEW

65

Records issue type, labels, state, and outcome on history entries, records what each PR-loop station did, tracks dispatch time, context, and stage hops, carries dispatch context onto waiting and parked records, and adds the underlying dispatch context columns via persistence migration 027.

— Migration named but not directly operator-actionable v1.28.0

11 **`dispatch.partials` shared template config key** NEW

65

Adds the `dispatch.partials` config key to share template partials across rules in a dispatch workflow.

— Named config key, but no usage example given v1.27.0

12 **Label-gated dispatch on GitHub, Gitea, and GitLab** BREAKING

60

Restricts dispatch to labeled issues only on GitHub, Gitea, and GitLab — issues without a matching label are no longer picked up by a running Sortie instance.

— Clear before/after but no migration command given v1.27.0

13 **Dispatch rule matching by title and per-rule agent settings** NEW

60

Adds title-phrase matching to dispatch rules so issues can be routed to an agent kind based on their title, and lets a dispatch rule carry its agent kind's own settings, enabling per-rule agent configuration.

— Mechanism described in prose without a config example v1.26.0

14 **Runtime safety: credential checks, secret redaction, early-exit reporting** IMPROVED

60

Verifies runtime credentials before the first turn on every agent kind, adds secret redaction from output when reporting early runtime exits, and reports early exits for every runtime kind, not only handshake kinds.

— Explains before/after scope but no configurable surface v1.25.0

THINNER COVERAGE BELOW

15 **Notification routing for Discord, Telegram, and per-destination subscriptions**

NEW 55

Adds Discord and Telegram notification kinds for alerting on run events, and routes outbound messages through per-destination subscriptions so each destination can carry its own routing.

– Names channels and routing concept but no config key v1.30.0 v1.26.0

16 **Literal comment posting across tracker adapters** NEW

55

Adds literal comment posting to every tracker adapter: GitHub Issues, GitLab Issues, Gitea Issues, Linear, and Jira.

– Lists all five adapters but no API or flag detail v1.26.0

17 **Live state streaming over server-sent events** NEW 45

Streams state changes over server-sent events so connected clients receive live updates without polling.

– Mechanism named but no endpoint or client example v1.28.0

18 **`protocolpin` schema-pin drift comparator** NEW 45

Adds `protocolpin`, a schema pin drift comparator for detecting Agent Client Protocol schema drift.

– Names the tool but not how to invoke it v1.25.0

19 **Per-run model recording (configured vs reported)** NEW

40

Records the configured model and the model actually reported by the agent for every run.

– Bare description with no surfacing location named v1.26.0

20 **Qualification framework expansion** IMPROVED 40

Adds a product conformance verdict, security, cleanup, identity, and end-to-end checks, continuation identity and token inventory, a live observation admission model, provenance artifact and measurement composition, semantic case inducers and asking launch, token inventory reading and grading, and session-continuation induction and grading, and scopes not-inducible grades per surface in qualification reports.

– Many named terms with no explained mechanism or usage v1.25.0

21 **Nightly CI results posted to Discord as embeds** NEW

40

Reports nightly CI outcomes and new issues to Discord as embeds.

— States what happens but no config or trigger detail v1.25.0

---

22 **Runtime profile schema v4** IMPROVED 20

Moves the runtime profile to schema v4.

— Bare version bump with no further detail v1.25.0

┗━▷ BREAKING ON UPGRADE

! Dispatch now only processes issues that carry a recognized label on GitHub, Gitea, and GitLab; issues without a matching label are skipped and will no longer be picked up by a running Sortie instance.

! The `kiro` agent kind is deprecated in favor of `agent-client-protocol`; existing workflows using `agent.kind: kiro` should migrate to `agent.kind: agent-client-protocol`.

WAS THIS USEFUL? 👍 👎

truefoundry TrueForge

Feedback ⓘ

@truefoundry/trueforge-ui@0.180.0 NOTES ↗ RANK

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

> TrueForge's 0.179–0.180 releases remove legacy schema-migration and internal import APIs in favor of the Migrator, add fine-grained Anthropic tool streaming and sandbox environment sharing with duplicate-name rejection, and round out session sharing, cancelled-turn visibility, and agent metrics across the UI.

└──▷ **WHAT SHIPPED · 18 FEATURES** most completely described first ? what's the number?

01 Cancelled turn and failed sub-agent visibility IMPROVED

78

Agent Sessions timeline and metrics now surface cancelled turn reasons and failed sub-agents; cancelled turns display via an orange cancelled banner in chat and session detail, projected from `turn.done` message metadata (including live commits), alongside MCP auth resume turn boundaries and unanswered `ask_user` prompts shown in the Sessions view.

— Mechanism and data source named across two releases @truefoundry/trueforge-ui@0.179.0 @truefoundry/trueforge-ui@0.180.0

02 Sandbox environment sharing and duplicate-name handling

75

BREAKING

Tenant members can now use sandbox environments created by other members, with update and delete restricted to the creator. The Environments UI is always enabled following removal of the `sandbox-environments` frontend feature flag, and creating a sandbox environment with a duplicate name via `POST` now returns a rejection error instead of silently overwriting the existing entry via upsert — callers relying on upsert semantics must handle the new rejection.

— Names flag, HTTP method and behavior change across releases

@truefoundry/trueforge-ui@0.180.0 @truefoundry/trueforge-ui@0.179.0 @truefoundry/trueforge@0.179.0

03 Fine-grained tool streaming for Anthropic models NEW

71

Fine-grained tool streaming is enabled via the TrueFoundry `anthropic-beta` header and Anthropic's `toolStreaming` option, and invalid tool-argument JSON is now reported as a distinct parse error.

– Names exact header and option but limited reader action

`@truefoundry/trueforge-core@0.180.0`

04 **Removed public-schema auto-migration env var** BREAKING

70

The `AUTOMATICALLY_MOVE_TRUEFORGE_TABLES_FROM_PUBLIC_TO_TRUEFORGE_SCHEMA` environment variable and the Postgres public-schema bootstrap are removed; schema management is now solely handled by the Migrator and init migration, and deployments relying on the old bootstrap must migrate to it.

– Names exact env var and required migration path `@truefoundry/trueforge@0.179.0`

05 **Analytics event namespacing and SDK timeout telemetry** IMPROVED

68

All public `AnalyticsEvents` names are now prefixed with `trueforge.` so host sinks (e.g. PostHog) can namespace product events, and every SDK HTTP timeout emits `trueforge.sdk.http_timeout` with the request endpoint, surfacing a `Request timed out` toast instead of a generic error.

– Names exact event keys but no reader-facing action beyond integration

`@truefoundry/trueforge-ui@0.180.0`

06 **Reasoning effort defaults in chat and agent config** IMPROVED

65

Reasoning effort now defaults to `medium` when the selected model offers it, with fallback order `medium` → `high` → `low`, applied in both the New Chat composer and the Agent Config builder.

– Clear behavior and two named UI surfaces to check `@truefoundry/trueforge-ui@0.180.0`

07 **Session sharing access controls** NEW

60

The session share URL is hidden when access is set to `Only you`, and shared-session viewers are routed to a read-only session view with in-flight turns hidden.

– Explains behavior but no exact UI path given `@truefoundry/trueforge-ui@0.180.0`

08 **Host-overridable Register Skills action** NEW

60

A host-overridable Register Skills action appears next to the Skills modal title, routing to Settings → Skills or a platform URL.

– Exact UI location given, mechanism thin `@truefoundry/trueforge-ui@0.180.0`

09 **Blocked large tool response diagnostics** IMPROVED

60

Blocked large tool response messages now state the exact content size and the character limit that blocked it, and the file-dump message prompts the model to plan

reads against the total.

— Explains new message content, no exact limit value given

`@truefoundry/trueforge-core@0.180.0`

THINNER COVERAGE BELOW

10 **Agents library usage metrics** NEW 55

Agents library rows now show per-page sessions, spend, a 14-day sparkline, and last-run metrics when available.

— Concrete metrics named but no exact navigation `@truefoundry/trueforge-ui@0.180.0`

11 **Model/MCP request resilience and timeout improvements** IMPROVED 55

SSRF fetch retry logic now covers `502` and `503` status codes across model, MCP, and outbound fetch operations, model and MCP failures are prefixed in logs and chat errors to identify which system failed, and default model HTTP headers and body timeouts are raised to 2 minutes to accommodate slow first-token and idle stream responses.

— Named status codes, limited direct user action Concrete new value, but no config key to adjust it

`@truefoundry/trueforge-core@0.180.0` `@truefoundry/trueforge@0.179.0`

12 **Web Search built-in connector toggle** NEW 50

New Chat now shows Web Search as a Built-in connector toggle, persisting the user's explicit draft choice across sessions and reloads.

— Clear UI starting point, limited mechanism detail `@truefoundry/trueforge-ui@0.179.0`

13 **External markdown link and image confirmation** NEW 50

Opening external markdown links or loading external markdown images now prompts a confirmation, with a per-host Don't ask again choice stored in localStorage.

— Names storage mechanism and behavior clearly `@truefoundry/trueforge-ui@0.179.0`

14 **Removed internal ServiceFoundry import API** BREAKING 50

The internal ServiceFoundry agent and session import API, including the unused managed-agent ID passed on agent create, has been removed; integrations targeting those internal endpoints will break.

— Names API scope but no specific endpoint paths given `@truefoundry/trueforge@0.179.0`

15 **Full-pane error for inaccessible session deep links** IMPROVED 35

Inaccessible session deep links now show a full-pane `Unable to load session` error instead of a toast-and-redirect.

— Simple before/after, no named surfaces `@truefoundry/trueforge-ui@0.180.0`

16 **Model settings dropdown access in Agent Config** IMPROVED 35

The model settings dropdown can now be opened by clicking the model parameters summary in Agent Config.

— Exact click target but trivial change `@truefoundry/trueforge-ui@0.179.0`

17 **Session turn count explanation** IMPROVED 30

The UI now explains higher session-level turn counts when the tip lineage is shorter than the session total.

— Vague description of a minor UI clarification `@truefoundry/trueforge-ui@0.179.0`

18 **Session creator avatar in sessions list** NEW 25

The sessions list meta row now shows the session creator's avatar, with the subject's name on hover.

— Minor cosmetic change, thinly described `@truefoundry/trueforge-ui@0.180.0`

NEW INTEGRATION You.com remote MCP server preset

UPDATED INTEGRATION Shipped model catalog updated with newer presets

└──▷ BREAKING ON UPGRADE

! The `AUTOMATICALLY_MOVE_TRUEFORGE_TABLES_FROM_PUBLIC_TO_TRUEFORGE_SCHEMA` environment variable and the Postgres public-schema bootstrap are removed; any deployment relying on them must migrate to using the Migrator and init migration for schema management.

! The internal ServiceFoundry agent and session import API, including the unused managed-agent ID passed on agent create, has been removed; any integration targeting those internal endpoints will break.

! Sandbox environment `POST` create now rejects duplicate names with an error instead of overwriting the existing entry via upsert; callers that relied on upsert semantics must handle the new rejection.

`@truefoundry/trueforge@0.180.0` NOTES ↗ STABLE

Adds Prometheus metrics at `GET /metrics`, per-request skill/MCP/model header overrides, tenant sandbox sharing, and structured log IDs.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch @truefoundry/trueforge@0.180.0
https://github.com/truefoundry/trueforge.git
# already have the repo? check out this version:
$ git checkout @truefoundry/trueforge@0.180.0
```

Exposes Prometheus request metrics at `GET /metrics` so a scraper can track rate, status, and latency for `/api` requests.

Adds `x-tfg-skills`, `x-tfg-mcp`, and `x-tfg-models` headers to inline session create, spec-replace, and turn-run routes, letting callers override skills, MCP servers, and models per-request without saving the overrides; `x-tfg-skills` requires git skills (registry skill IDs are rejected), and MCP servers and models fall back to the registry when a name is missing.

Uses `rediss://` URL scheme for standalone Redis when `REDIS_TLS_ENABLED` is `true`, advertising the secure protocol in the connection URL.

Adds optional AgentSpec-shaped `model` field (`{ name: provider/model, params? }`) on `model.message` events, stamped from the bound agent definition for new turns.

Adds tenant, request, session, and turn IDs to server and controller logs so failures can be filtered to a single request or turn.

+5 more

WAS THIS USEFUL? 👍 👎

LangGenius Dify

Feedback ⓘ 1.18.0 NOTES ↗ RANK

Dify is an LLM application platform for building agentic workflows and RAG pipelines through a visual workspace, deployable to cloud or self-hosted.

> Dify 1.18.0 ships a new difyctl CLI, portable .ifpkg app packages, Agent Output Routes for branch selection, nested workflow trace drilldown, and a beta Agent Knowledge Base, alongside several breaking API, CLI and runtime changes requiring coordinated upgrades.

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

01 **difyctl CLI for workflow and skill management** NEW 95

Adds `difyctl` CLI with server-discovered commands supporting Workflow and Chatflow creation, DSL import/export, settings, environment variables, publishing, version restoration, and agent skill installation via `difyctl install skills <dir>`. The v2 client does not reuse v1 login state, so a fresh `difyctl login` is required after upgrading.

Authenticate the new difyctl v2 client and browse available server-discovered commands after upgrading to 1.18.0.

```
$ difyctl login --server https://your-dify.example.com
difyctl help
```

Install bundled agent skills into a local Dify agent from a directory, enabling agents to build and manage Dify apps via the CLI.

```
$ difyctl install skills <dir>
```

— Names many commands and flags with runnable examples 1.18.0

02 **Mandatory auth token for Agent sandboxes** BREAKING 85

Local Agent sandboxes now require a non-empty authentication token; the `1.17.1` runtime image does not support the new `--require-auth-token` flag and must be replaced with the `1.18.0` image.

— Names exact flag and the image versions involved 1.18.0

03 **Composition v2 required on /runs endpoint** BREAKING 75

The `/runs` endpoint now accepts composition v2 and rejects v1 requests; old Agent runtimes cannot read newly saved v2 snapshots and must be upgraded together with API, workers, Agent Backend, and Agent runtime images.

— Explains upgrade scope but no version numbers given `1.18.0`

04 **Per-mode OpenAPI routes replace unified run endpoint** BREAKING 75

Replaces the old mode-agnostic `/openapi/v1/apps/{app_id}:run` route with per-mode operations discoverable through the server's operation catalog; hard-coded v1 routes must be updated to use the new catalog.

— Names exact deprecated endpoint but no catalog example `1.18.0`

05 **.ifpkg portable app packages** NEW 70

Introduces `.ifpkg` portable app packages that bundle Workflow and Chatflow apps with inline Agents, Skills, files, icons, and Site settings into a single exportable artifact; imports accept `.ifpkg` or YAML from files or URLs.

— Describes bundle contents with a screenshot, no CLI command `1.18.0`

06 **Nested workflow tool trace drilldown** NEW 70

Adds nested Workflow Tool trace drilldown: open a Workflow Tool invocation from run history and inspect the child workflow's individual steps, with coordinated pause, resume, stop, and file outputs.

— Explains navigation and controls with a screenshot `1.18.0`

07 **Agent Knowledge Base (Beta)** NEW 70

Introduces Agent Knowledge Base (Beta), a new KnowledgeFS-backed knowledge space with local upload preview, connected-source management, document availability and metadata, retrieval history, and quality evaluation alongside classic Knowledge.

— Names the KnowledgeFS backend and capabilities, no exact path `1.18.0`

08 **Updated source install requirements** BREAKING 70

Source installations now require Python 3.13 for the API, Node.js 24.22.1 or later in the 24.x line, and pnpm 12.10.1.

— Gives exact version numbers for each toolchain `1.18.0`

09 **Agent Output Routes for branch selection** NEW 60

Adds Agent Output Routes, allowing an Agent to select the next workflow branch via a read-only `switch` output and configurable selection conditions.

.

— Names the switch output but lacks condition detail `1.18.0`

---

10 **Knowledge Retrieval v2 node and knowledge bindings** NEW

60

Adds Knowledge Retrieval v2 node and Agent knowledge bindings using scoped capabilities; published Explore runs reuse published bindings without requiring viewers to reconfigure access.

— Names the node and binding behavior, no config keys `1.18.0`

THINNER COVERAGE BELOW

11 **AskHuman tool and deferred resume removed** BREAKING

45

The old Agent-specific AskHuman tool and deferred resume path are removed: historical Agent forms no longer resume execution.

— States removal but gives no migration path `1.18.0`

---

12 **Streaming output for Agent v2 nodes** IMPROVED

30

Enables plain Agent v2 nodes to stream text output as it is generated.

— Thin one-line description with no mechanism `1.18.0`

└──▷ BREAKING ON UPGRADE

! The `/runs` endpoint now accepts composition v2 and rejects v1 requests; old Agent runtimes cannot read newly saved v2 snapshots and must be upgraded together with API, workers, Agent Backend, and Agent runtime images.

! The old Agent-specific AskHuman tool and deferred resume path are removed: historical Agent forms no longer resume execution.

! The old mode-agnostic `/openapi/v1/apps/{app_id}:run` route is replaced by per-mode operations; hard-coded v1 routes must be updated to use the server-discovered catalog.

! The `difyctl` v2 client does not reuse v1 login state; a fresh `difyctl login` is required after upgrading.

! Local Agent sandboxes now require a non-empty authentication token; the `1.17.1` runtime image does not support the new `--require-auth-token` flag and must be replaced with the `1.18.0` image.

! Source installations now require Python 3.13 for the API, Node.js 24.22.1 or later in the 24.x line, and pnpm 12.10.1.

WAS THIS USEFUL? 👍 👎

Langflow

Feedback ⓘ v1.13.0 NOTES ↗ RANK

Langflow is a visual builder for agents and RAG workflows that exports each flow as an API or Python object for deployment.

> Langflow v1.13.0 ships a new triggers framework (webhooks, schedules, Slack/Microsoft/Google events), managed OAuth connections for integrations, and a suite of CLI and API tooling for migrating, inspecting, and administering self-hosted instances, alongside several smaller enterprise administration and deployment features.

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

01 **Migration tooling: CLI commands, API endpoints, and guided Settings page**

NEW 90

Adds a full migration toolkit: five CLI commands with `--json` output (`check-integrity`, `migration-preflight`, which can accept a target Langflow version, `convert-sqlite-to-postgres`, `relocate-files`, which copies uploaded file bytes into S3 with a bucket pre-check, and `relocate-kb`), migration API endpoints to pause/resume writes, test and save the destination, confirm the secret key, back up the instance, run database copy, and run knowledge base and file copies, plus a guided `Settings > Migration` page for superusers covering the same destination, secret key, pause, backup, database copy, and knowledge base/file copy steps.

Before migrating to a new Langflow instance, verify the source instance is internally consistent and the migration can succeed.

```
$ langflow check-integrity --json
langflow migration-preflight --json
```

Move uploaded files to S3 and relocate knowledge base vectors to a new backend as part of a storage migration.

```
$ langflow relocate-files --json
langflow relocate-kb --json
```

Convert a self-hosted SQLite database to Postgres and get machine-readable output to pipe into a migration script.

```
$ langflow convert-sqlite-to-postgres --json
```

— Names all five commands and flag, with runnable examples. Named UI path and endpoint purposes but no exact routes.

v1.13.0

---

02 **Triggers framework with webhook, schedule, and event-source triggers**

NEW

80

Adds a trigger entity, event ledger, triggers API, leased dispatcher, retries, and dead-lettering as core trigger infrastructure, plus a schedule trigger, an inbound webhook trigger with signed push ingress, Slack message and reaction triggers via the Events API and Socket Mode, and Microsoft and Google source events for triggers.

— Mechanism named (dispatcher, retries, dead-lettering) but no exact API paths. v1.13.0

---

03 **Managed OAuth connections and integration bundles** NEW

65

Adds a persistent connection API and delegated OAuth connection brokering for integrations, with connection-ref fields rendered via a connection picker in the UI, plus an API to list OAuth registrations a caller may authorize. Ships Microsoft and Slack connection-backed bundles and Google Workspace actions on managed connections.

— Names integrations and UI picker but no concrete endpoint paths. v1.13.0

THINNER COVERAGE BELOW

04 **Atomic replacement and snapshots for deployments** NEW

50

Adds atomic replacement and snapshots for deployments, including required model providers and models declared in deploy artifacts and MCP server provisioning.

— Explains scope but no API or config names given. v1.13.0

---

05 **Team, user, and directory administration via API and CLI** NEW

45

Adds team and user administration API and CLI, and enterprise directory provenance tracking with its own CLI.

— Names API and CLI surfaces but no commands or endpoints given. v1.13.0

---

06 **Pre-creation hook registry for projects, users, roles** NEW

30

Adds an enterprise pre-creation hook registry for projects, users, and roles.

— Names scope but no mechanism or configuration detail. v1.13.0

07 **Per-resource access exceptions for authorization** NEW

25

Adds per-resource access exceptions for authorization.

— Bare one-line description with no mechanism given. v1.13.0

08 **Application audit log for flows and projects** NEW

25

Adds an application audit log for flows and projects.

— Bare description with no fields or retrieval mechanism given. v1.13.0

09 **Opt-in retention for terminal jobs** NEW

20

Adds opt-in retention for terminal jobs.

— Single-line description with no configuration or limits given. v1.13.0

WAS THIS USEFUL? 👍 👎

assistant-ui

Feedback ⓘ `@assistant-ui/react@0.15.31` NOTES ↗

RANK

Assistant-ui provides React components and state management for chat interfaces with streaming, tool calls, and attachments.

> assistant-ui's latest release adds fine-grained composer text-change tracking and improves thread-list unarchive handling.

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

01 **Composer text-change event tracking** NEW 75

A new `composer.textChange` store event fires on every composer text write, carrying the resulting text and an `origin` (e.g. clear, send, reset, history recall) describing why the draft changed. `setText` is extended to accept `{ origin }` so programmatic text updates record their source in the emitted event.

— Names exact event and param but no usage example `@assistant-ui/react@0.15.31`

THINNER COVERAGE BELOW

02 **Thread-list unarchive support** NEW 40

Thread-list now exposes unarchive support, and unarchive controls are automatically disabled when the underlying adapter cannot restore archived threads.

— Describes behavior but no API/flag named `@assistant-ui/react@0.15.31`

WAS THIS USEFUL? 👍 👎

AWS Strands Agents

Feedback ⓘ harness-cli/v0.2.2 NOTES ↗

RANK

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

> Strands Agents' harness-cli v0.2.2 release adds agent.cancel() for stopping runs mid-execution, makes the middleware system public, enforces invoke limits across agent trees, and introduces a new swarm tool for multi-agent coordination alongside TypeScript tracing support and several catalog additions.

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

01 **agent.cancel() for stopping agent runs** NEW 75

Adds agent.cancel() to stop the Python agent event loop mid-run, letting a caller interrupt a long-running invocation such as on a user interrupt or timeout.

Cancel a long-running Python agent mid-flight, e.g. on a user interrupt or timeout.

python

```python
import asyncio
from strands_harness import create_harness

agent = create_harness()

async def run():
    task = asyncio.create_task(agent.invoke_async('Analyse every
log in /var/log'))
    await asyncio.sleep(5)
    agent.cancel()   # stop the event loop
    await task

asyncio.run(run())
```

— Named function with a runnable cancellation example. harness-cli/v0.2.2

THINNER COVERAGE BELOW

02 **New skills, provider, and plugin in catalogs** NEW 50

Adds `docs-verifier` skill, `attenu-guard` (community intervention) skill, `opencode` model provider, and `strands-you` (You.com web tools plugin) to the catalogs.

— Four named additions, no usage instructions included. harness-cli/v0.2.2

03 **Invoke-limit enforcement across agent trees** NEW 45

Introduces a new invocation-limit mechanism that enforces invoke limits and usage accounting across the whole agent tree.

— Mechanism named but no API or config surface shown. harness-cli/v0.2.2

04 **Baggage propagation ported to TypeScript** NEW 45

Ports baggage propagation to TypeScript, enabling distributed-tracing context to flow across agent hops.

— Describes behaviour but no code sample given. harness-cli/v0.2.2

05 **Public Python middleware system** IMPROVED 40

Makes the Python middleware system public, enabling production use of request/response interception hooks.

— No code example or hook signature given. harness-cli/v0.2.2

06 **Swarm tool for multi-agent coordination** NEW 40

Adds a swarm tool for multi-agent coordination, available in both the Python and TypeScript SDKs.

— No usage example or tool signature provided. harness-cli/v0.2.2

07 **Harness, CLI, and box release streams** NEW 25

Adds harness, CLI, and box release streams to the changelog system.

— Thin description of internal changelog tooling. harness-cli/v0.2.2

08 **Experimental OpenAI Live provider** NEW 25

Adds an experimental OpenAI Live model provider.

— Only named in the release summary, no further detail. harness-cli/v0.2.2

WAS THIS USEFUL?  

Feynman

Feedback ⓘ v0.5.53 NOTES ↗ RANK

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

> Feynman's Pi 1.4.0 update adds a new environment variable to control concurrency of parallel research subagents.

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

01 **Configurable codemode worker concurrency** NEW 75

The `PI_CODEMODE_MAX_WORKERS` environment variable raises the concurrency limit for in-process parallel research subagents waiting on codemode workers, e.g. `PI_CODEMODE_MAX_WORKERS=6 feynman deepresearch "mechanistic interpretability"`.

Allow more parallel codemode workers when running deep research workflows with many simultaneous subagents.

```
$ PI_CODEMODE_MAX_WORKERS=6 feynman deepresearch "mechanistic interpretability"
```

— Names exact env var and runnable example command. v0.5.53

WAS THIS USEFUL? 👍 👎

openclacky

Feedback ⓘ v1.5.21 NOTES ↗ LATEST STABLE

The most Token-efficient open-source AI Agent

v1.5.21 adds live Manual/Auto approval switching mid-task, a configurable `ask_user_timeout_seconds` in `~/.clacky/config.yml`, configurable PID file path, multiline terminal command support, bulk session archiving, and a dependency-free PNG256 fallback encoder. The file reader per-line limit rises to 8000 characters and oversized browser screenshots are now downscaled to cut token cost.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.5.21 https://github.com/clacky-ai/openclacky.git
# already have the repo? check out this version:
$ git checkout v1.5.21
```

Adds `ask_user_timeout_seconds` to `~/.clacky/config.yml` to set how long auto-approve mode waits on `ask_user` before the agent continues (default 20 seconds; `0` continues immediately).

Switch the current session between Manual and Auto approval from the composer, even while a task is running.

Configure where the server writes its PID file so you can run multiple instances or meet system path requirements.

Supports multiline terminal commands.

Falls back to a dependency-free PNG256 encoder when no image tool is available.

+3 more

NEW INTEGRATION DeepSeek hosted web search and page fetch

NEW SUPPORTED PLATFORM MiniMax M3.1 Flash model

RETIRED SUPPORTED PLATFORM GPT-6 and Grok 4.7 aliases removed from model picker

NEW INTEGRATION New model aliases: oc-grok-4.7, abs-claude-haiku-5-5, or-gemini-4-argon

RETIRED INTEGRATION Retired gemini-3.7-flash model and oc-minimax-m2.7 alias; removed OrcaRouter and Requesty presets

NEW INTEGRATION QQ official-bot channel

NEW INTEGRATION Seedance reference image forwarding via OpenAI-compatible video gateway

NEW SUPPORTED PLATFORM oc-glm-5.3 and oc-kimi-k3 model aliases

NEW SUPPORTED PLATFORM abs-claude-sonnet-5-5 and abs-gpt-6.1-sol model aliases

NEW SUPPORTED PLATFORM Xiaomi MiMo V2.6 models

NEW SUPPORTED PLATFORM oc-minimax-m2.7 alias

└──▷ BREAKING ON UPGRADE

! The GPT-6 and Grok 4.7 aliases are removed from the model picker; existing sessions routed to those lanes continue to work, but new sessions cannot select them by name.

▾ 2 more releases in this issue · 2026-10-10 → 2026-10-22

v1.5.20 NOTES ↗ STABLE

Adds session archiving with recycle bin, audio/video/image previews, zoom/pan in image viewer, per-task feedback countdowns, and auto session naming.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.5.20 https://github.com/clacky-ai/openclacky.git
# already have the repo? check out this version:
$ git checkout v1.5.20
```

Adds a data management page with session archiving and a recycle bin.

Supports audio and video file preview in the workspace.

Supports preview of pending image attachments before sending.

Adds zoom and pan controls in the viewer's image preview.

Adds opt-in per-task feedback countdowns.

+5 more

└──▷ BREAKING ON UPGRADE

! The `gemini-3.7-flash` model and the `oc-minimax-m2.7` alias are removed from the model picker — any session or config referencing them will need to be updated.

! The OrcaRouter and Requesty provider presets are removed — configurations using those presets must be migrated to a different provider.

v1.5.19 NOTES ↗ STABLE

Adds session lifecycle hooks, chat navigator bookmarks, drag-to-move sessions, reworked billing heatmap, and an `Add Model` picker button.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.5.19 https://github.com/clacky-ai/openclacky.git
# already have the repo? check out this version:
$ git checkout v1.5.19
```

Adds session lifecycle hooks for programmatic control over session state transitions.

Adds bookmarks to the chat navigator with a tightened layout.

Supports dragging a session onto a project to move it.

Reworks the billing charts with a year heatmap and a 365-day range.

Adds an `Add Model` button to the model picker.

+2 more

WAS THIS USEFUL?  

lexmount Moli

Feedback ⓘ v1.1.19 NOTES ↗ RANK

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

> Moli expanded its web-platform rendering surface with native WebAuthn credential handling, searchable PDF rendering, and an instance-scoped `Browser.close` over CDP, alongside broad native interface coverage spanning CSS, WebRTC, Canvas, SVG, IndexedDB, forms, fetch timing and the WebIDL binding engine.

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

01 **Native WebAuthn credential support** NEW 65

Adds native WebAuthn support: parses JSON credential options with native WebIDL conversion and binds native `CredentialsContainer` entry points.

— Names the entry point but no usage example given v1.1.17

02 **Instance-scoped Browser.close via CDP** NEW 65

Adds `Browser.close` shutdown scoped to individual browser instances via CDP.

— Names a callable API and its scoping mechanism v1.1.19

03 **Expanded native Web Platform API coverage across subsystems** NEW 65

Adds native PictureInPicture event payloads for media and speech synthesis events with validated payloads, along with expanded native Web Platform API coverage: BMP image format support via the shared raster codec and native `ReadableStream.from` iteration (r0); `CSSAnimation` / `CSSTransition` interface objects, opaque self-signed certificate primitives in `crypto`, non-constructible transport and RTP interfaces in `webrtc`, `MediaSource` handle and buffer interface shells, native payment update event values, native `BarProp` objects and popup visibility in `window`, and preserved path winding rules in native snapshots via `paint` (r1); a native streaming XHR response decoder for encoding (r2); and `ImageBitmapRenderingContext` in Canvas, missing CSS properties with CSS-wide keywords via CSSOM, native UTF-16 name values in IndexedDB, missing native Window and DOM event handlers, button Auto state and input value attribute modes in the forms core, blob URL snapshotting with creator access keys in the File API, and precise classification of `about:srcdoc` URLs (r3).

— Names two subsystems with no usage path shown Names every exposed interface across many subsystems

v1.1.17 v1.1.16 v1.1.18 v1.1.19

04 **Native SVG interfaces and rendering fixes** IMPROVED 65

Exposes native SVG value and element interfaces, honors fill rules and rejects nonfinite fill points, retains transform functions and rotation centers, and exposes point-list parsing with incomplete-pair handling in SVG.

— Detailed behavior but no direct invocation surface v1.1.16

05 **WebIDL conversion and binding improvements** IMPROVED

65

Adds clamped long long conversion and derive support (r0); fixes the non-shared `BufferSource` converter (r1); derives wrapping signed long long conversion, sequences of dictionary values, octet arguments with modulo-256 conversion, `EnforceRange` unsigned short conversion, writes `BufferSource` rows without changing padding, and supports lossless `DOMString16` (r2); and adds unrestricted binary32 float conversion, `EnforceRange` signed long long support, derived dictionary member order and inheritance, retained native `BufferSource` identity through conversion, and lossless native DOM string values (r3).

— Enumerates every named conversion but no user-facing call v1.1.16 v1.1.17 v1.1.18 v1.1.19

THINNER COVERAGE BELOW

06 **Performance subsystem worker timing sharing** IMPROVED

50

Improves the `performance` subsystem by sharing native User Timing and observers with workers and exposing the remaining timing interface objects.

— Explains mechanism but no direct API call given v1.1.18

07 **Searchable PDF rendering** NEW

45

Renders searchable PDFs with hyperlink preservation, verified in CI.

— Describes outcome, no API or flag to invoke it v1.1.17

08 **Fetch subsystem timing and request handling** IMPROVED

45

Preserves native response timing observations and observes native request and interim response timing for `fetch` requests; isolates subrequest cancellation and response progress in the fetch subsystem; and adds explicit interaction support before page dump in fetch.

— Groups fetch internals improvements without a usage example v1.1.16 v1.1.18 v1.1.19

09 **CSP policy engine extracted into moli-csp** IMPROVED

40

Extracts the shared CSP policy engine into `moli-csp`.

— Names the crate but not what changed for users v1.1.19

10 **Taffy and Stylo layout engine upgrade** IMPROVED 40

Upgrades the Taffy and Stylo layout engines with sizing and overflow improvements.

— Names engines but improvements described vaguely v1.1.19

WAS THIS USEFUL? 👍 👎

ag2 Feedback ⓘ v1.2.1 NOTES ↗ RANK

AG2 (formerly AutoGen): The Open-Source AgentOS.Join us at: https://discord.gg/sNGSwQME3x

> ag2 shipped OpenAI Decisions API support, Mistral speech-to-text/text-to-speech integration, new search and scraping toolkits, and A2A server replica support, alongside several breaking changes affecting toolkit composition, ACP multimodal handling, Mistral typing, and MCP tool routing.

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

01 MistralConfig async client retyped to httpx2 BREAKING

75

`MistralConfig.async_client` is now typed as `httpx2.AsyncClient`; `httpx.AsyncClient` still works at runtime but fails type-checking, and `httpx.HTTPError` no longer catches `httpx2` errors. Requires `mistralai>=3.2.0,<4`.

— Clear before/after mechanism but no migration code. v1.2.1

02 OpenAI Decisions API support NEW

75

Adds `OpenAIDecisionsConfig` and shared decision schemas via the `Question` marker for OpenAI Decisions API support.

Use the OpenAI Decisions API to route structured agent decisions with a shared schema marker.

python

```python
from ag2.config import OpenAIDecisionsConfig
from ag2 import Agent
from ag2.decisions import Question

config = OpenAIDecisionsConfig(model="gpt-4o")
agent = Agent("decider", prompt="You make routing decisions.",
config=config)

# Question marker signals which fields are decision points in
the shared schema
class RouteDecision(Question):
    destination: str
    confidence: float
```

— Runnable example configures a decisions agent directly. v1.2.0

03 **Ollama provider API key and HTTP client config** NEW

75

Adds API key, custom HTTP client, and tool name configuration to the Ollama provider, e.g. OllamaConfig(model="llama3", api_key="ollama-secret", http_client=httpx.AsyncClient(verify=False)).

Connect ag2 to a local Ollama instance that requires an API key and a custom HTTP client.

python

```python
import httpx
from ag2 import Agent
from ag2.config import OllamaConfig

agent = Agent(
    "local",
    prompt="You are a helpful assistant.",
    config=OllamaConfig(
        model="llama3",
        api_key="ollama-secret",
        http_client=httpx.AsyncClient(verify=False),
    ),
)
```

— Runnable example shows full Ollama configuration. v1.2.0

04 **Reasoning effort control in OpenAIResponsesConfig** NEW

70

Adds a `reasoning` option to `OpenAIResponsesConfig` for controlling effort and summaries, e.g. OpenAIResponsesConfig(model="o3", reasoning={"effort": "high"}).

Ask an OpenAI Responses agent to reason with explicit effort control.

python

```python
from ag2.config import OpenAIResponsesConfig

config = OpenAIResponsesConfig(model="o3", reasoning={"effort": "high"})
```

— Runnable example sets reasoning effort directly. v1.2.1

05 **Multimodal content blocks preserved in ACP and network layer**

70

BREAKING

ACP and the network layer now preserve multimodal input parts as separate content blocks instead of flattening them to text; `ChannelAdapter.render_envelope` and `extract_turn_input` can now return an `Input` or a sequence of them. Custom channel adapters that relied on a single joined string must be updated to handle `Input` or a sequence of them.

— Clear before/after but no migration code sample. v1.2.1

06 **Toolkit composition middleware inheritance change** BREAKING 70

Changes `left | right` toolkit composition so that the right-hand toolkit's tools no longer inherit middleware from the left, and later additions to the combined toolkit inherit neither source's middleware; code relying on composition to propagate middleware must now attach it explicitly.

— Clear before/after mechanism with migration note. v1.2.0

07 **Mistral voice support in ag2.live** NEW 65

Adds `MistralTranscriber` for offline speech-to-text and `MistralRealtimeTranscriber` for realtime speech-to-text, plus `MistralTTSConfig` and `MistralStreamingTTSConfig` for text-to-speech, all in `ag2.live`, requiring `mistralai>=3.2.0,<4`.

— Names four new classes but no usage example given. v1.2.1

08 **MCP tool calls pinned to discovery server** BREAKING 60

MCP tool calls are now pinned to the configuration used for discovery; a Variable(default_factory=...) can no longer route a call to a different server than the one that advertised the tool.

— Explains behavior change, no migration example. v1.2.1

THINNER COVERAGE BELOW

09 **Compaction verifier for CompactStrategy** NEW 55

Adds the `compaction_verifier` extension to measure whether a `CompactStrategy` degrades an agent's next actions, beyond just token savings.

— Explains purpose but no usage example. v1.2.1

10 **A2A server validation and replica support** NEW 55

Adds configurable push notification URL validation to A2A servers, and supports running `A2AServer` on multiple replicas with cacheable agent cards and opt-in input media type validation.

— Names A2AServer features but no usage example. v1.2.0

11 **Bounding-box image queries in TypeSafe** NEW 50

Enables ResponseSchema(BoundingBox) in TypeSafe to locate objects in images by sending a Box question.

— Named schema but no code example shown. v1.2.1

12 **UnsupportedInputError for cross-provider files** BREAKING 50

OpenAI Chat Completions now raises `UnsupportedInputError` when files uploaded to another provider are included in messages.

— Names new exception, no handling example. v1.2.1

13 **New search and scraping toolkits** NEW 45

Adds `FirecrawlToolkit` for web search and scraping via the Firecrawl API, and `KeenableSearchToolkit` for Keenable-backed search.

— Two toolkits named, no usage example given. v1.2.1

14 **ElevenLabs dependency version bump** BREAKING 35

The ElevenLabs extra now requires `elevenlabs>=2.73.0,<3`.

— Bare dependency bump, no further detail. v1.2.1

15 **Explicit API keys in OpenAI voice configs** IMPROVED 25

OpenAI voice configs now accept explicit API keys.

— Single-line addition with no further detail. v1.2.0

╰──▷ BREAKING ON UPGRADE

! `MistralConfig.async_client` is now typed as `httpx2.AsyncClient`; `httpx.AsyncClient` still works at runtime but fails type-checking, and `httpx.HTTPError` no longer catches `httpx2` errors. Requires `mistralai>=3.2.0,<4`.

! ElevenLabs extra now requires `elevenlabs>=2.73.0,<3`.

! ACP and the network layer no longer join input parts into a single string — each part is sent as its own content block. Custom channel adapters that rely on `ChannelAdapter.render_envelope` or `extract_turn_input` must be updated to handle `Input` or a sequence of them.

! MCP tool calls are now pinned to the configuration used for discovery; a Variable(default_factory=...) can no longer route a call to a different server than the one that advertised the tool.

! OpenAI Chat Completions now raises `UnsupportedInputError` when files uploaded to another provider are included in messages.

! `left | right` toolkit composition no longer attaches the left toolkit's middleware to tools from the right, and later additions to the combined toolkit do not inherit either source's middleware — code that relied on composition to propagate middleware must now attach it explicitly.

WAS THIS USEFUL? 👍 👎

openclaw.net

Feedback ⓘ v0.4.2 NOTES ↗ RANK

Self-hosted Enterprise and Personal AI + agent runtime in .NET (NativeAOT-friendly)

> `OpenClaw.NET` v0.4.2 ships breaking access-control changes that now require an `operator` role for agent execution and mutating APIs, alongside new Nacos and Vault/OpenBao provider resolution for capability and secret lookup, per-call delegated MCP credentials and protocol improvements, Fractal Memory workflows, Jev/Laya decision routing, audience controls and remote device enrollment.

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

01 Operator role required for execution and chat

BREAKING

90

Owned sessions now accept writes only from their owner or an administrator. The `operator` role is now required for agent execution, `Companion` / `CLI` chat, and mutating APIs, with newly created operator-account records defaulting to the `viewer` role — existing accounts will lose access to those surfaces until granted `operator`. The deprecated `OpenClaw:Security:AllowViewerAgentExecution` setting (JSON/CLI) and `OpenClaw__Security__AllowViewerAgentExecution` environment variable must be removed entirely, even if set to `false`, or the gateway refuses to start.

— Names exact role, config keys and startup failure condition v0.4.2

02 Nacos capability provider and default change

BREAKING

80

Adds a `provider: nacos` config key and an explicit Nacos adapter build for capability resolution; the default provider is now `local`, with no Nacos SDK dependency in the standard gateway. Existing Nacos integrations must switch to the explicit adapter build and set `provider: nacos` in configuration to keep working.

Switch an existing Nacos capability integration to the explicit adapter after upgrading, so capability resolution continues to work with your Nacos service registry.

```yaml
provider: nacos
```

— Exact config key, build variant and migration example given v0.4.2

03 Vault/OpenBao secret resolution for JIT gateways

NEW

75

Adds Vault/OpenBao secret resolution (opt-in) in JIT gateway builds via a `vault:` reference scheme. The distributed NativeAOT gateway excludes the VaultSharp backend and fails closed on `vault:` references, so Vault resolution requires a JIT build.

— Names reference scheme and build-specific fail-closed behaviour v0.4.2

04 **MCP delegated credentials and protocol improvements** IMPROVED 73

Adds per-call delegated credentials for MCP tool invocations, meta-skill invocation idempotency for MCP workflows, session metadata forwarding in MCP app tool calls, and structured MCP responses with improved protocol/bridge failure handling; MCP packages are aligned at version `2.2.0`.

— Lists several named MCP changes and pinned package version v0.4.2

05 **Capability execution reliability improvements** IMPROVED 70

Improves capability execution with binding trajectory evidence and retry/fallback handling across native and Microsoft Agent Framework runtimes, plus bounded binding caches for shared local capability resolution.

— Names runtimes and caching mechanism but no config surface v0.4.2

THINNER COVERAGE BELOW

06 **Audience controls for model context and tools** NEW 50

Adds audience controls for model context and tool dispatch; enabling the policy requires fresh sessions, since active sessions will only pick up the new policy after renewal.

— Describes behaviour but no config key or API named v0.4.2

07 **Remote device enrollment and managed updates** NEW 50

Adds remote device enrollment and managed-update primitives; managed-update rollback can select older binaries but does not downgrade persisted data.

— States rollback limitation but no concrete commands given v0.4.2

08 **Jev/Laya decision routing tooling** NEW 40

Adds hosted Jev turn-routing evaluation and local Laya decision routing tooling, both opt-in; the release explicitly notes this does not establish production routing quality.

— Named but caveated as non-production, no usage path given v0.4.2

09 **Fractal Memory capture and resume workflows** NEW 25

Adds optional Fractal Memory capture and resume workflows for agent sessions.

— Bare description with no mechanism or usage detail v0.4.2

╰──▷ BREAKING ON UPGRADE

- ! The `operator` role is now required for agent execution, Companion/CLI chat, and mutating APIs; newly created operator-account records default to `viewer` and will lose access to those surfaces until granted `operator`.
- ! `OpenClaw:Security:AllowViewerAgentExecution` (JSON/CLI) and `OpenClaw__Security__AllowViewerAgentExecution` (environment variable) must be removed — even when set to `false` — or the gateway refuses to start.
- ! The default capability provider changes to `local`; existing Nacos integrations must switch to the explicit Nacos adapter build and set `provider: nacos` in configuration.
- ! Vault/OpenBao `vault:` secret references fail closed in the distributed NativeAOT gateway build, which excludes the VaultSharp backend; use JIT gateway builds for Vault resolution.
- ! Owned sessions now reject writes from any account that is not the session owner or an administrator.
- ! Audience restrictions require fresh sessions after the policy is enabled; enabling them with active sessions will apply the new policy only to new sessions.
- ! Managed-update rollback selects older binaries but does not downgrade persisted data.

WAS THIS USEFUL? 👍 👎

yoagent

Feedback ⓘ v0.25.12 NOTES ↗ RANK

The agent loop for Rust — stream from 7 LLM protocols, run tools, loop until done. From a laptop to a Cloudflare Worker.

> yoagent shipped an extension plug-in framework that hooks the full agent lifecycle, a decision-model system with tool gating and input guarding, completed Amazon Bedrock support with SigV4 signing and streaming, added wasm32/Cloudflare Workers build support, and overhauled model pricing with a runtime-overridable price table, new priced presets, and cost-accounting fixes across providers.

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

01 Provider retry handling: overloaded retries, retry events, and retry-safe streams

IMPROVED 96

Overloaded providers are retried on HTTP 529/503 and in-stream signals (`overloaded_error` , `UNAVAILABLE`), logging the server explanation. A retried attempt emits `AgentEvent::ProviderRetry` (wire tag `providerRetry`) carrying `attempt` , `max_attempts` , `error` , and `delay_ms` , following a closing `MessageEnd` with `StopReason::Error` and an `error_message` — consumers must check whether the next event is `ProviderRetry` before treating an error `MessageEnd` as final. retry::retry_safe_events(rx) / `retry::RetrySafeEvents` hold back each attempt's events until it succeeds, preventing partial text from a retried attempt from reaching the consumer; a final failed attempt keeps its start and end events but not its deltas, making it suitable for non-interactive, append-only output such as stdout on a pipe or a log.

Wrap an agent's event receiver so retried provider attempts never emit partial text to a downstream pipe or log.

rust

```
use yoagent::retry;

let mut events =
retry::retry_safe_events(agent.prompt("Summarise src/").await);

while let Some(event) = events.recv().await {
    // partial deltas from failed attempts are suppressed;
    // only events from a successful attempt reach here
    if let AgentEvent::MessageUpdate { delta: StreamDelta::Text
{ delta }, .. } = event {
        print!("{delta}");
    }
}
```

Observe retry events to log backoff details or surface them in a dashboard without treating them as final failures.

```rust
while let Some(event) = events.recv().await {
    match event {
        AgentEvent::ProviderRetry { attempt, max_attempts,
error, delay_ms } => {
            eprintln!("[retry {attempt}/{max_attempts}] waiting
{delay_ms}ms — {error}");
        }
        AgentEvent::AgentEnd { .. } => break,
        _ => {}
    }
}
```

— Full event schema and two runnable examples across three releases. v0.24.3 v0.24.1 v0.23.0

---

02 **Decision models with tool gating and input guarding** NEW

96

Adds DecisionModel::jev(), jev_opencode(), local(url), and logprobs(url, model) constructors for typed decision queries (yes/no, Choice, Score) against text or JSON state, behind the `decision` feature flag, with fallback chains via .or(backup) sharing a single deadline across the chain, and `calibrate` to measure decision model accuracy, Brier score, ECE, thresholds, and fitted temperature over a labelled dataset. `Agent::with_decision_model` attaches an advisory decision model that hints on skills and tools (failing open on errors); Agent::with_tool_gate(ToolGate::new(model)) denies tool calls judged destructive and not user-requested (failing closed); with_input_guard(InputGuard::new(model)) screens agent input for prompt injection or harmful content (failing closed by default). Decision model spend is recorded in `SessionStats::decision` and included in `total_cost_usd`. Also adds `ToolCallRequest::new` builders for unit-testing middleware, `AsyncInputFilter` and `TurnHook` extension points, `SubAgentTool::with_turn_hook` / `with_async_input_filter`, and a structured user_request_parts() accessor on hooks.

Gate destructive tool calls at runtime using a self-hosted decision model, so the agent cannot execute them unless the user explicitly requested it.

```rust
let model = DecisionModel::local("http://localhost:8080");
let agent = Agent::from_config(ModelConfig::claude_sonnet_5())
    .with_tools(tools::default_tools())
    .with_tool_gate(ToolGate::new(model));
```

Screen every incoming user message for prompt injection using an OpenAI-compatible open-weight model that returns logprobs, failing closed on any detection.

```rust
let guard_model =
DecisionModel::logprobs("http://localhost:1234/v1", "mistral-
7b");
let agent = Agent::from_config(ModelConfig::claude_sonnet_5())
    .with_tools(tools::default_tools())
    .with_input_guard(InputGuard::new(guard_model));
```

Chain a primary hosted decision model with a local fallback under one deadline, so agent evaluation continues even if the primary is unavailable.

```rust
let primary = DecisionModel::jev();
let backup = DecisionModel::local("http://localhost:8080");
let model = primary.or(backup);
let agent = Agent::from_config(ModelConfig::claude_sonnet_5())
    .with_decision_model(model);
```

— Multiple named constructors, fail-open/closed semantics, three examples. v0.21.0

03 Price table with runtime overrides, live fetch, and re-pricing

96

BREAKING

`src/provider/prices.json` holds every built-in model price with its source URL and check date, loaded and validated by `PriceTable`. The `YOAGENT_PRICES` environment variable points at a local `prices.json` to replace built-in prices without a code change; prices::install_override(table) installs overrides programmatically (returning a report of changed, reverted, and inapplicable entries); env_override_status() / load_env_override() let an application fail fast on a bad price file. PriceTable::fetch_cached(source, path, CacheOptions) fetches and disk-caches prices from a live source (reporting origin, age, and errors), supporting `PriceSource::YoagentMain`, `PriceSource::ModelsDev`, and custom `Url` sources, plus `InstallPolicy::AddOnly` so fetched prices only fill in unlisted models. ModelConfig::reprice(), Agent::reprice(), and SubAgentTool::reprice() re-price configs built before a price table was installed. Precedence is three-level:

runtime override beats fetched prices beats the built-in file, while a manually set `config.cost` beats all three (replaced only by reprice() or with_prices()). A one-time `warn!` per model fires when a first-party price lookup misses while only `YOAGENT_PRICES` / `install_override` / `install_fetched` is active (no bundled snapshot), and `LivePrices` is marked `#[must_use]`. Breaking: first-party constructors (`anthropic` , `openai` , `openai_responses` , `google` , `xai` , `groq` , `deepseek` , `mistral` , `zai` , `minimax` , `qwen` , `meta`) now return Some(cost) instead of `None` for listed models, and meta(id, name) now prices per model id, returning `None` for ids not in the price list.

Override model prices at startup without shipping a new release — useful when a provider changes rates between yoagent versions.

```
$ YOAGENT_PRICES=/etc/yoagent/prices.json cargo run --release
```

Fetch the latest prices from the yoagent main branch and cache them to disk, so merged price fixes arrive without a library upgrade.

```rust
use yoagent::provider::prices::{PriceSource, PriceTable,
CacheOptions};

let (table, report) = PriceTable::fetch_cached(
    PriceSource::YoagentMain,
    "/var/cache/yoagent/prices.json",
    CacheOptions::default(),
).await?;
prices::global::install_override(table);
```

Re-price an agent config that was constructed before a custom price table was installed, ensuring cost tracking reflects the new table.

```rust
let mut agent =
Agent::from_config(ModelConfig::anthropic("claude-opus-5-5",
"Claude Opus"))
    .with_system_prompt("You are a helpful assistant.");
// Install override first, then re-price configs built before
it.
prices::global::install_override(table);
let agent = agent.reprice();
```

— Config file path, env var, methods and precedence all named, with examples. v0.25.0

v0.20.0

04 ToolGate full-view guarantees for large arguments

BREAKING

93

`with_max_argument_chars` (default 48,000) sets the character budget at which ToolGate sends arguments whole instead of truncating them. with_full_view(tool, json_pointer) marks specific arguments — such as `bash`'s `/command` — as requiring a full view or denial, never a partial cut, and `without_default_full_views` removes the built-in full-view protection for `bash`'s `/command`. Cut metadata is exposed under `tool_call.omitted` so the decision model knows it is judging a partial view, and partial views are logged at `info` level listing which arguments were cut and how many characters the gate did not see, alongside every refusal. Breaking: a `bash` command too large to review whole (roughly 48,000 characters serialized, or the decision backend's token limit if smaller) is now denied outright instead of being judged from its head and tail.

Require the gate to judge `bash`'s full command or deny it, while raising the overall character budget to 64,000 for other tools.

```rust
let gate = ToolGate::new(backend)
    .with_max_argument_chars(64_000)
    .with_full_view("bash", "/command");
```

Remove the default full-view requirement for `bash` when your decision backend has a tight token limit and you accept partial reviews.

```rust
let gate = ToolGate::new(backend)
    .without_default_full_views();
```

— Exact flags, default value, and breaking behaviour named. v0.25.4

05 wasm32/Cloudflare Workers support via native feature

BREAKING

93

Adds `yoagent::rt` with `spawn`, `sleep`, `timeout`, `JoinHandle`, and `Instant` — backed by Tokio on native targets and by the host executor / `setTimeout` / performance.now() on `wasm32-unknown-unknown`. A new default `native` feature gates filesystem and shell tools, default_tools(), `StdioTransport` / `with_mcp_server_stdio`, `FileBackend`, and `PriceTable::fetch_cached`; `cargo build --target wasm32-unknown-unknown --no-default-features` builds the agent loop, all providers, HTTP MCP, `SharedState`, sub-agents, compaction, and the `decision` feature for Cloudflare Workers and other wasm32 targets. Breaking: setting

`default-features = false` now also removes `native`, disabling those tools and reqwest's default features (including TLS), so every HTTPS provider call fails at runtime unless `features = ["native"]` is added alongside.

Build yoagent for Cloudflare Workers or any wasm32 target without pulling in native-only dependencies.

```
$ cargo build --target wasm32-unknown-unknown --no-default-features
```

— Runnable build command and exact migration fix given. v0.23.0

---

06 **New priced model presets across providers** NEW SUPPORTED PLATFORM

UPDATED SUPPORTED PLATFORM NEW 90

Adds eleven priced model preset constructors — claude_sonnet_5_5(), claude_haiku_5_5(), gpt_6_1_sol(), gemini_4_argon(), gemini_3_8_flash(), grok_4_7(), mistral_large_4(), glm_5_3(), glm_5_3_flash(), qwen_3_8_max(), qwen_3_8_flash() — each carrying the vendor's real context window and output-token limit, activated after prices::enable_bundled(). Also adds bundled pricing for `deepseek-flash`, `deepseek-v4-pro`, and the legacy `deepseek-v4-flash` (billed as Flash), sourced from DeepSeek's pricing page; `CostConfig` and `Budget` enforce peak rates, with off-peak (half price outside 01:00–04:00 and 06:00–10:00 UTC on weekdays) requiring a manual override per the pricing guide. Live-pricing support via `prices::enable_live*` installs models.dev's current promotional rate rather than the snapshot upper bound for `gemini_4_argon`, `gemini_3_8_flash`, and `mistral_large_4`.

Use Grok 4.7 with correct reasoning-cost accounting out of the box — no extra setup needed beyond enabling bundled prices.

```rust
use yoagent::prices;
use yoagent::provider::ModelConfig;
use yoagent::{Agent, tools};

#[tokio::main]
async fn main() {
    prices::enable_bundled();
    let mut agent = Agent::from_config(ModelConfig::grok_4_7())
        .with_tools(tools::default_tools());
    // grok_4_7() carries the 200K-context tier and counts
reasoning tokens in output
    let mut events = agent.prompt("Refactor src/main.rs for
clarity").await;
    // ... handle events
    agent.finish().await;
}
```

Enable bundled pricing so DeepSeek model costs are tracked against peak rates before constructing your agent.

```rust
use yoagent::prices;

prices::enable_bundled();
// deepseek-flash, deepseek-v4-pro, and deepseek-v4-flash are
now priced
// CostConfig and Budget will enforce peak-rate upper bounds
```

— Names every constructor and pricing rule across two releases. v0.25.8 v0.25.6

07 Cloudflare Clef decision models and Workers AI integration NEW

90

DecisionModel::clef(account_id) and DecisionModel::clef_flash(account_id) (feature `decision`) call `@cf/cloudflare/clef` and `@cf/cloudflare/clef-flash` on Workers AI via the SystemOne API, reading the token from `CLOUDFLARE_API_TOKEN` then `CLOUDFLARE_AUTH_TOKEN`. SystemOneBackend::workers_ai(account_id, model_path) and SystemOneBackend::with_endpoint_url(url) are lower-level constructors with automatic envelope unwrapping and retry logic (code 3040 retried, 3036 not). A new crate `yoagent-workers` (`integrations/yoagent-workers/`) exposes `yoagent_workers::ai::{clef, clef_flash, AiBackend}` to run Clef through the Workers AI binding (`env.AI`) as a `DecisionBackend` — no API token or account ID required; `AiBackend::with_retry` configures retry behavior. decision::parse_systemone_response(body, request) is now public for backends that

move requests themselves, and `DecisionError::RateLimited` gains a `body` field (variant `#[non_exhaustive]`) plus a `rate_limited_with_body` constructor to surface the server's raw response body.

Implement a custom backend that moves requests itself and reuses yoagent's SystemOne parser for Cloudflare envelope unwrapping.

```rust
use yoagent::decision::parse_systemone_response;

let body = my_transport.send(&request).await?;
let decision = parse_systemone_response(&body, &request)?;
```

— New crate, endpoint model names, and retry codes all specified. v0.24.0

08 **Extension plug-in contract for the agent loop** NEW 88

`Agent::with_extension` / `SubAgentTool::with_extension` installs an Extension (in `yoagent::extension`) that hooks the full lifecycle via `RunHooks`: `tools`, `on_input`, `before_model`, `before_tool`, `after_tool`, `on_stop`, `on_event`, and `finish` (with RunOutcome::end() returning a `RunEnd`), plus a defaulted `RunHooks::take_failure` so an extension can hand over a failure recorded in `on_event` or background work. `with_tree_extension` propagates an extension to every delegated sub-agent run at any depth, exposed via `Agent::delegated_from` / `AgentLoopConfig::delegated_from` and `ToolContext::{tree_extensions, delegation_depth, run_label}`. Also adds `Agent::with_run_label` and `Agent::with_max_stop_continues` for run configuration, six runnable offline extension examples (`extension_policy`, `extension_redact`, `extension_verifier`, `extension_budget`, `extension_tree`, `extension_audit`, runnable via `MockProvider` or `-- --live` with `DEEPSEEK_API_KEY` / `ANTHROPIC_API_KEY`), and `tests/extension_combos_test.rs` covering extensions combined with tool strategies, compaction, steering, `prompt_structured`, provider retries, and middleware.

Attach a tool-call policy that denies dangerous commands — the extension runs for every sub-agent in the tree, so host policy is enforced at any delegation depth.

```rust
use yoagent::extension::ClonedHooks;
use yoagent::{ToolCallRequest, ToolDecision};

#[derive(Clone)]
struct NoRmRf;

#[async_trait::async_trait]
impl yoagent::extension::RunHooks for NoRmRf {
    async fn before_tool(&self, call: &ToolCallRequest<'_>) ->
ToolDecision {
        let cmd =
call.args["command"].as_str().unwrap_or_default();
        if cmd.contains("rm -rf") {
            ToolDecision::Deny("rm -rf is not allowed".into())
        } else {
            ToolDecision::Allow
        }
    }
}

let agent = Agent::from_config(ModelConfig::claude_sonnet_5())
    .with_tools(tools::default_tools())
    .with_tree_extension(ClonedHooks::new("no-rm-rf", NoRmRf));
```

— Full lifecycle hooks and examples named; no numeric limits given. v0.25.0

09 **OpenAI Decisions API backend** NEW SUPPORTED PLATFORM NEW

87

`OpenAiDecisionBackend` (feature `decision`) targets `POST /v1/decisions`, configured via `with_api_key` / `with_base_url` / `with_retry` / `with_safety_identifier` / `with_capabilities`, with preset DecisionModel::gpt_6_luna() and custom models via `with_model`, priced from the bundled entry `openai-decisions/gpt-6-luna` at $0.10 per million input tokens with output free. `DecisionError::Refused { message }` (plus `DecisionError::refused`, `is_refusal`) terminates the fallback chain immediately via `DecisionModel::or`, so the tool gate and input guard fail closed.

— Exact endpoint, config methods, and pricing named. v0.25.0

10 **Amazon Bedrock support completed with SigV4 and streaming**

87

BREAKING

Adds `AWS_BEARER_TOKEN_BEDROCK` for Bedrock API key authentication alongside `AWS_*` IAM credential variables, resolved per-request, plus an `api_key` field on

`ModelConfig` accepting `access:secret[:token]` format or a Bedrock API key for explicit credential supply. Full Amazon Bedrock streaming via `ConverseStream` is decoded as AWS event-stream frames with CRC checks — a dropped or corrupt stream is an error, never a silent empty turn — and SigV4 request signing for Bedrock is verified against AWS published test vectors. Breaking: Bedrock IAM credentials are now signed by yoagent directly; users of a prior signing proxy must set `AWS_REGION` or add a placeholder `authorization` header in `ModelConfig.headers` to avoid double-signing.

— Exact env vars, field format, and migration step named. `v0.22.0`

---

11 **Search tool hardening against config leakage and silent failures**

85

BREAKING

The `search` tool now exposes `details.engine` reporting `"rg"` or `"grep"` so callers can inspect which binary ran, locates `rg` without relying on `which` (so it correctly selects ripgrep instead of silently falling back to `grep` in minimal containers), and runs `rg` with `--no-config` while clearing `RIPGREP_CONFIG_PATH` and `GREP_OPTIONS` so user or environment config cannot corrupt path-scoped output or escape `SearchTool::with_allowed_paths` through symlinks. Abnormal search exits (signal, unexpected exit code, read error) now surface as an error or a 'results may be incomplete' warning rather than silently reporting `No matches`, and the tool now returns partial results when some files cannot be read, surfacing unreadable-file errors as warnings instead of discarding matches. Breaking: clearing `RIPGREP_CONFIG_PATH` / `GREP_OPTIONS` means prior ripgrep/grep config no longer affects built-in search results (it still applies to `rg` / `grep` invoked through `bash`).

— Names every env var and field changed; breaking scope spelled out. `v0.25.5` `v0.24.3`

---

12 **Per-request header injection for HTTP MCP servers** NEW

82

HttpTransport::with_header(name, value) sends a header — auth token, API key, `User-Agent` — on every request of an HTTP MCP session, connected via McpClient::connect_http_with(transport) or Agent::with_mcp_server_http_transport(transport), including on wasm32 / Cloudflare Workers.

Authenticate every request to a bearer-token-protected HTTP MCP server without hardcoding headers in each call.

```rust
use yoagent::mcp::HttpTransport;
use yoagent::Agent;

let transport = HttpTransport::new("https://mcp.example.com")
    .with_header("Authorization", "Bearer <token>");

let agent = Agent::from_config(config)
    .with_mcp_server_http_transport(transport);
```

— Exact method signature and two connection paths, with runnable example. v0.25.2

13 **Spend tracking for extensions** NEW 80

`RunContext::spend_handle() -> Option<SpendHandle>` lets extensions record model calls into a run's budget and stats, via SpendHandle::add(&usage, &model) and SpendHandle::add_priced(&usage, Option<&CostConfig>) from any task while the run is live. Extension-originated spend appears in a new `SessionStats::extensions` bucket (`ExtensionSpend { usage, cost_usd, requests }`), folds into total_usage() / total_cost_usd() and sub-agents' runs, and Agent::extension_spend() sums it across all runs.

Bill a model call made inside an extension to the current run so it counts against the Budget and appears in SessionStats.

```rust
if let Some(handle) = ctx.spend_handle() {
    handle.add(&usage, &model);
}
```

— Names every API and struct field; runnable snippet included. v0.25.10

14 **ToolSource trait for dynamic tool sets** NEW 79

`ToolSource` (`yoagent::ToolSource` , module `tool_source`) defines one async method `tools() -> Vec<Arc<dyn AgentTool>>` , installed via `Agent::with_tool_source` / `SubAgentTool::with_tool_source` , consulted once per run and dropped at run end — enabling plugin systems or reconnecting MCP servers to supply tools without restarting the agent.

Attach a dynamic tool set that changes at runtime — e.g. a plugin system or reconnecting MCP server — without restarting the agent.

```rust
use std::sync::Arc;
use yoagent::{Agent, AgentTool, ToolSource};
use yoagent::provider::ModelConfig;

struct MyPluginSource;

#[async_trait::async_trait]
impl ToolSource for MyPluginSource {
    async fn tools(&self) -> Vec<Arc<dyn AgentTool>> {
        load_plugins_from_registry().await  // your dynamic
loader
    }
}

let agent = Agent::from_config(ModelConfig::claude_sonnet_5())
    .with_tool_source(Arc::new(MyPluginSource));

let mut events = agent.prompt("List available actions").await;
```

— Exact trait signature and install methods, with a runnable example. v0.23.0

15 **Enumerate all credential environment variables** NEW

79

provider::api_key_variables() returns every environment variable yoagent reads credentials from — per-provider key vars (including Bedrock's `AWS_*`), the `YOAGENT_API_KEY` / `API_KEY` fallbacks, and decision presets' keys — for hosts that need to blank credentials before starting plugin runtimes or MCP servers.

Enumerate all credential env vars before spawning an MCP server so none leak into the child process.

```rust
use yoagent::provider;

let key_vars = provider::api_key_variables();
let mut cmd = std::process::Command::new("mcp-server");
for var in &key_vars {
    cmd.env_remove(var);
}
cmd.spawn().expect("mcp-server started cleanly");
```

— Exact function and use case, with runnable example. v0.25.12

16 **Multi-block tool results fully delivered to the model** IMPROVED

78

All text blocks from a multi-block tool result now reach the model — Anthropic, OpenAI-compatible, Responses (OpenAI and Azure), Gemini, and Vertex providers previously sent only the first block and dropped the rest. Blocks are joined with a newline via the crate-private `provider::tool_result_text` path, omitting empty blocks, and per-block `max_lines` truncation is applied independently so a tool answering in many blocks can contribute more total content than a single block could. Array form is preserved for results with images on Anthropic, OpenAI-compatible, and Responses providers, and `inlineData` parts on Gemini and Vertex, when mixing text and image blocks. The GASP recorder's tool output summary now reads every block, recording (no output) for an empty result.

— Explains mechanism across all 7 providers; no runnable example given. v0.25.9

---

17 **yoagent-rutis plugin bridge for rutis and pi-ecosystem extensions**

NEW

78

New crate `yoagent-rutis` (`integrations/yoagent-rutis/`) bridges rutis plugins into a live `Agent` — contributing tools via `ToolSource` , gating tool calls, adding turn notes, rejecting input, and observing `AgentEvent` s on the rutis bus. Version 0.1.1 adds pi-ecosystem extension bridging: tools, tool policies, input checks, prompt additions, and images load unchanged through an adapter using pi's own loader, failing closed when a pi feature cannot be enforced. Images in tool results are now supported across TypeScript, Python, DSH, and rutis-agent runtimes.

— Names the crate, version, and runtimes, but no code sample. v0.25.1 v0.23.0

---

18 **Encrypted reasoning carry-through for OpenAI/Azure** NEW

76

ModelConfig::with_encrypted_reasoning(true) (`OpenAiCompat::encrypted_reasoning`) carries reasoning across tool calls for OpenAI Responses and Azure OpenAI providers — opt-in, off by default. It sends `include: ["reasoning.encrypted_content"]` and replays reasoning items with their paired output item ids (`fc_…` / `msg_…`).

— Exact flag, request field, and id formats named. v0.25.0

---

19 **Model preset iteration and exact lookup** NEW

75

`ModelConfig::presets() -> impl Iterator<Item = ModelConfig>` iterates all 22 named model presets. `ModelConfig::preset(provider, id) -> Option<ModelConfig>` does exact, case-sensitive lookup without building unmatched presets, and `ModelConfig::generic(provider, id) -> Option<ModelConfig>` constructs a provider's generic model config by name, routing OpenAI reasoning ids to the Responses API and returning `None` for gateways, local servers, and custom endpoints.

Resolve a user-supplied `provider/model-id` string to a fully-configured ModelConfig, falling back to the generic constructor when no preset exists.

```rust
let config = ModelConfig::preset("xai", "grok-4.7")
    .or_else(|| ModelConfig::generic("xai", "grok-4.7"))
    .expect("unknown provider or model");
```

— Three named lookup functions with exact return behaviour. v0.25.10

---

20 **Thinking/reasoning token cost accounting fixes**

UPDATED SUPPORTED PLATFORM IMPROVED 75

Gemini and Vertex `Usage::output` now includes thinking tokens (`thoughtsTokenCount`), correcting cost accounting and ensuring the context tracker and `max_total_tokens` limit fire on thinking-heavy turns. OpenAI-compatible providers now add `completion_tokens_details.reasoning_tokens` to `output` when those tokens are reported beside (not inside) `completion_tokens` , correctly pricing models such as Grok 4.7 whose reasoning is always on.

— Names the exact fields fixed and models affected. v0.25.8

---

21 **GASP recorder captures rutis plugin logs** NEW 72

GaspRecorder::extension() and GaspRecorder::plugin_log_layer() record a rutis plugin's `log` lines as `observation.created` events on the run they belong to, routed to the host's `tracing` span under the key `yoagent.log` and tagged with the run id. Enabling the `gasp` feature in `Cargo.toml` now also pulls in `tracing-subscriber` (registry only).

— Named functions and tracing key, no example given. v0.25.1

---

22 **Compaction cost tracking and re-pricing** NEW 72

LlmCompaction::reprice(&self) and a defaulted `CompactionStrategy::reprice` extend `Agent::reprice` to also re-price the compaction summarizer. `SessionStats::compaction` (`CompactionSpend { usage, cost_usd, requests }`) and Agent::compaction_spend() expose what `LlmCompaction` summarization requests cost.

— Named structs and methods but no usage example. v0.25.0

---

23 **MCP over stdio reliability fixes** IMPROVED 69

MCP over stdio now matches responses by id, treats `initialized` as a real notification, drains stderr with the tail and exit status recorded on server death, honours cancellation and timeout on calls, and no longer fails a call on unknown content types.

— Five concrete behaviour fixes, no exact API surface named. v0.24.3

24 **Configurable sub-agent execution limits** IMPROVED 67

`SubAgentTool::with_execution_limits` makes sub-agent token and duration caps configurable instead of hard-coded.

Cap a sub-agent's token budget and wall-clock time so a runaway parallel reviewer cannot exhaust quota.

```rust
use yoagent::sub_agent::SubAgentTool;

let reviewer = SubAgentTool::new("security-reviewer",
agent_config)
    .with_execution_limits(max_tokens, max_duration);
```

— Exact method named with a runnable example. v0.24.3

25 **Anthropic config inferred from model id** IMPROVED 64

ModelConfig::anthropic("claude-...") now infers thinking mode and native structured outputs from the model id, instead of requiring manual configuration.

Let yoagent pick thinking mode and structured-output settings automatically from a Claude model id instead of configuring them manually.

```rust
use yoagent::provider::ModelConfig;

// thinking mode and native structured outputs inferred from the
id
let config = ModelConfig::anthropic("claude-sonnet-5", "Claude
Sonnet 5");
```

— Exact function named with a runnable example. v0.24.3

26 **File tools refuse non-regular files** IMPROVED 62

`read_file`, `write_file`, and `edit_file` now refuse non-regular files (FIFOs, character/block devices, sockets, directories) after following symlinks, returning an error that names the file type.

— Three named tools and the exact file types rejected. v0.25.8

27 **Anthropic redacted_thinking block support** NEW 62

Supports Anthropic `redacted_thinking` blocks, which are preserved and sent back unmodified as required when thinking is used with tools, via new `redacted` and `redacted_protocol` fields on the `Content::Thinking` variant.

— Named fields and variant, no worked example. v0.22.0

28 **BashTool kills entire process group on timeout, cancel, or drop** 61

BREAKING

On Unix, `BashTool` now kills the entire process group on timeout, cancel, or drop — pipeline stages, `&&` lists, and background jobs no longer keep running. The terminal's Ctrl+C no longer reaches a running command; hosts must cancel or drop the call instead.

— Clear before/after behaviour but no migration API given. v0.25.2

29 **bash tool exits as soon as the shell exits** IMPROVED 60

`bash` now exits as soon as the shell process exits rather than waiting for background jobs to close their output pipes; a `NOTE:` line and `details.output_still_open` field are set when a background job still holds the pipes.

— Names the exact field surfaced for the edge case. v0.25.8

THINNER COVERAGE BELOW

30 **prompt_cache_key for OpenAI Responses and Azure OpenAI** NEW 59

Adds `prompt_cache_key` on the OpenAI Responses and Azure OpenAI APIs, using `CacheConfig::session_key` or a system-prompt-derived key when caching hints are on.

— Named field and config source, no worked example. v0.25.0

31 **Agent-readable docs shipped in the crate** NEW 55

Adds `skills/yoagent/SKILL.md` in AgentSkills format to the crate, giving agents building on yoagent an in-registry skill reference with feature guides, recipes, tips, and per-version changelogs, plus `llms.txt` as a navigable map of every docs page for agents that browse documentation rather than read it directly.

— Names both files but no usage mechanics beyond that. v0.25.3

32 **Failed streams reported as errors, not truncated successes** 53

IMPROVED

Failed streams are now treated as errors rather than truncated finished responses, covering OpenAI-compatible mid-stream errors, Gemini streams with no finish reason, blocked prompts, and malformed function calls.

— Lists the covered cases but no API or migration note. v0.24.3

33 **Compaction headroom floor raised** IMPROVED 52

Raises the compaction floor `MIN_HEADROOM_RATIO` from 0.15 to 0.30, preventing tool-heavy sessions from losing the original task prompt during compaction.

— Exact before/after values but no action required. v0.22.0

---

34 **Cancelled runs end with StopReason::Aborted** BREAKING

43

A cancelled run now ends with `StopReason::Aborted` instead of `StopReason::Error`, so `on_error` is no longer called for cancellations.

— Clear before/after but no migration mechanism given. v0.23.0

---

35 **Steering messages preserved through run cancellation** IMPROVED

40

Steering and follow-up messages sent during a run cancellation are now appended to history with their `MessageStart` / `MessageEnd` events before the cancel marker, so the next run receives them instead of losing them.

— Describes behaviour but no API surface to act on directly. v0.25.10

└──▷ BREAKING ON UPGRADE

! The `search` tool now clears `RIPGREP_CONFIG_PATH` and `GREP_OPTIONS` for searches run through the built-in tool; any ripgrep or grep configuration previously applied via those environment variables no longer affects built-in search results (though it still applies to `rg` or `grep` invoked through `bash`).

! A `bash` command too large to review whole (roughly 48,000 characters serialized, or the decision backend's token limit if smaller) is now denied outright instead of being judged from its head and tail. Agents relying on the old truncation behaviour for large `bash` scripts will see those calls refused.

! On Unix, `BashTool` now kills the entire process group on timeout, cancel, or drop — pipeline stages, `&&` lists, and background jobs no longer keep running. The terminal's Ctrl+C no longer reaches a running command; hosts must cancel or drop the call instead.

! Setting `default-features = false` now removes the new `native` feature, disabling filesystem and shell tools, default_tools(), `StdioTransport` / `with_mcp_server_stdio`, `FileBackend`, `PriceTable::fetch_cached`, and all of reqwest's default features (including TLS) — every HTTPS provider call fails at runtime. Migration: add `features = ["native"]` alongside `default-features = false`.

! A retried provider attempt now emits a closing `MessageEnd` with `StopReason::Error` and an `error_message`, followed by `AgentEvent::ProviderRetry { attempt, max_attempts, error, delay_ms }`. A consumer that treats an error `MessageEnd` as the turn's final failure must check whether the next event is `ProviderRetry`.

! Bedrock IAM credentials are now signed by yoagent directly; if you previously used a signing proxy, you must set `AWS_REGION` or add a placeholder `authorization` header in `ModelConfig.headers` to avoid double-signing.

! First-party constructors (`anthropic` , `openai` , `openai_responses` , `google` , `xai` , `groq` , `deepseek` , `mistral` , `zai` , `minimax` , `qwen` , `meta`) now return Some(cost) instead of `None` for listed models — code that expected `None` from these constructors will see a price.

! meta(id, name) now prices per model id; ids not in the price list return `None` instead of the previous single fixed price.

v0.19.0 NOTES ↗ STABLE

Adds ModelConfig::openai_responses() and `ThinkingLevel::XHigh` / `Max` variants; `ModelConfig::cost` becomes `Option<CostConfig>` ; `ThinkingLevel` and `AnthropicCompat` are now `#[non_exhaustive]` .

└──▷ GET THIS VERSION ▾

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

Adds ModelConfig::openai_responses() constructor for building configs that target the OpenAI Responses API, including pricing for requests over 272K input tokens.

Extends `ThinkingLevel` with `XHigh` and `Max` variants, routed only to models that declare support for them; each preset declares its highest effort level.

└──▷ BREAKING ON UPGRADE

! `ThinkingLevel` is now `#[non_exhaustive]` ; match arms outside the crate must add a wildcard arm or they will fail to compile.

! `ModelConfig::cost` is now `Option<CostConfig>` : `None` means pricing is unknown; a config with every rate at zero means the model is free. Code that assumed a non-optional value will break.

! `AnthropicCompat` is now `#[non_exhaustive]` ; struct literal construction no longer compiles — use its constructors and builders instead.

! Claude presets now use native structured outputs via `output_config.format` ; schemas must set `additionalProperties: false` and must not use `minimum` or `maxLength` .

WAS THIS USEFUL? 👍 👎

anton

Feedback ⓘ v2.26.10.21.1 NOTES ↗ RANK

self-improving agent

> Anton now surfaces per-turn memory token counting, giving visibility into how much learned context the model sees each conversation turn.

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

01 **Per-turn memory token counting** NEW 30

Anton counts the learned memory each conversation turn places in front of the model, giving visibility into per-turn memory token usage.

— Describes what it does but no mechanism, UI path, or output format. v2.26.10.21.1

NEW DETECTION write-check: overwrite, HTTP write, procedure call, SQLAlchemy, and data-store write detection

> v2.26.10.19.3 NOTES ↗ STABLE
>
> └──▷ **GET THIS VERSION** ▾
>
> ```
> $ git clone --branch v2.26.10.19.3
> https://github.com/mindsdb/anton.git
> # already have the repo? check out this version:
> $ git checkout v2.26.10.19.3
> ```

WAS THIS USEFUL? 👍 👎

AWS Strands Decider

Feedback ⓘ v0.2.0 NOTES ↗ RANK

A small, fast decision model, or system one model, for agentic workflows. Pick between options or rate on a scale faster than an LLM, with a calibrated confidence on every decision.

> Strands Decider v0.2.0 expands its model suite with five new v1 models and adds a checkpoint-continuation flag for training, while splitting training and inference CLI surfaces.

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

01 **init_from checkpoint continuation for training** NEW

75

Adds `init_from` option to the `train` command to continue training from an already-trained model checkpoint.

– Names exact flag and command, ready to use v0.2.0

02 **Evaluation path aligned with serving** IMPROVED

70

Evaluation now refuses `--device mlx` and prepares the model using the same path as serving, ensuring consistent behaviour between evaluation and inference.

– Names flag and exact behavioural change v0.2.0

THINNER COVERAGE BELOW

03 **New Gemma and Qwen3.5 models added** NEW

55

Adds four new Gemma v1 models (2B, 4B, 12B, and 24B parameters) and a new Qwen3.5 2B v1 model to the model suite.

– Names model sizes but gives no usage command v0.2.0

04 **Training commands split from inference CLI** BREAKING

50

Training commands are split from inference in the CLI; training commands are dropped from the inference CLI surface. Users who invoked training via the shared CLI entrypoint must now use the separate training CLI surface.

– Describes change but no exact new command path given v0.2.0

└──▷ **BREAKING ON UPGRADE**

! Training commands have been removed from the inference CLI; users who invoked training via the shared CLI entrypoint must now use the separate training CLI surface.

WAS THIS USEFUL? 👍 👎

AG-UI

Feedback ⓘ release/2026-10-19 NOTES ↗ RANK

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

> AG-UI's biggest changes this window are breaking: a `@ag-ui/mastra@2.0.0` overhaul of interrupts, tool approval, and capability reporting, plus wire-format and peer-dependency bumps across several adapters, alongside new capabilities like multimodal A2A file-part encoding, native deferred approvals in the Claude SDK adapters, `RUN_STARTED` protocol-version declarations, and Python 3.15 support in ag_ui_adk.

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

01 Langroid endpoint CORS and route configuration

BREAKING

83

The `ag-ui-langroid@0.2.0` endpoint helper gains an `origins` argument and forwards FastAPI route kwargs (`name` , `tags` , `dependencies` , ...) for flexible route configuration. CORS credentials are no longer enabled for the default wildcard origin; pass explicit origins via the `origins` argument if browsers call the agent with cookies or credentials from another origin.

— Gives exact argument name and migration instruction. release/2026-10-12

02 Mastra interrupt handling redesign

BREAKING

82

Interrupts in `@ag-ui/mastra@2.0.0` now flow exclusively through the `RUN_FINISHED` interrupt outcome and `RunAgentInput.resume` , replacing the legacy custom interrupt mechanism. This removes the legacy CUSTOM `on_interrupt` event, the `forwardedProps.command` resume input, and the `emitInterruptOutcome` option; only one interrupt per run is now emitted, whereas multi-pause runs previously reported all pauses.

— Names exact fields removed and the replacement, no migration snippet.

release/2026-10-12

03 Watsonx adapter endpoint and streaming fixes

IMPROVED

80

`@ag-ui/watsonx` and `ag_ui_watsonx` gain a `baseUrl` option to point the adapter at private endpoints or OpenAI-compatible mocks instead of the URL built from `region` and `instanceId` ; `ag_ui_watsonx` forwards FastAPI route options (`name` , `tags` , `dependencies`) through its endpoint helpers with FastAPI imported only on use; tool results with content parts now pass through to `TOOL_CALL_RESULT` ; and streamed tool

calls stay on the assistant message in `MESSAGES_SNAPSHOT` so frontend tools run correctly.

— Several named options but no single runnable example. release/2026-10-16

04 **Mastra tool approval validation** BREAKING 80

In `@ag-ui/mastra@2.0.0`, malformed tool approval answers now fail the run with `MASTRA_INVALID_TOOL_APPROVAL` instead of silently declining. Tool approval now accepts only `true` or `{ approved: true }` to approve, and declines only on cancel or `{ approved: false }`; any other answer fails the run.

— Names exact accepted values and the new error code. release/2026-10-12

05 **Deferred approvals and tool result changes in Claude SDK** 78

BREAKING

`@ag-ui/claude-agent-sdk` and `ag-ui-claude-sdk` now support native deferred approvals, letting workflows pause for human approval before continuing. This ships alongside breaking changes: in `@ag-ui/claude-agent-sdk@0.0.7` tool result serialization and the frontend payload format changed (re-verify tool names, result envelopes, and failure statuses), and in `ag-ui-claude-sdk@0.2.2` tool error reporting switched to the Python-native error flag and the structured tool result envelope/frontend payload serialization changed.

— Explains mechanism and migration notes but no code sample. release/2026-10-16

06 **Python 3.15 support in ag_ui_adk** BREAKING 75

`ag_ui_adk` 0.10.0 raises its `requires-python` upper bound to `<3.16` to support Python 3.15, and now requires `ag-ui-a2ui-toolkit 0.0.5` for Python 3.15 installs; the A2UI SDK dependency is pinned to `0.2.2`.

— Clear version pins but only a lockfile-check migration note. release/2026-10-18

07 **A2A middleware clone and debug logging improvements** IMPROVED 74

`@ag-ui/a2a-middleware@0.0.5` improves clone() to carry over agent cards, clients, instructions, and orchestrator state across per-request clones (e.g. CopilotKit runtimes), preventing first-run failures. `@ag-ui/a2a-middleware@0.0.6` adds `debug.enabled` gating to delegation logging so A2A task and reply logs are emitted via `console.debug` only when `debug.enabled` is true.

— Names exact method, flag and versions, no runnable command. release/2026-10-13

08 **maxProtocolVersion cap in Spring AI adapter** DEPRECATED 73

`@ag-ui/spring-ai` adds `maxProtocolVersion`, replacing the deprecated `maxVersion`, to cap the protocol at 0.0.39 for compatibility with the community Java SDK, which is not yet on AG-UI 1.0.

— Names exact fields and version ceiling, narrow scope. release/2026-10-13

09 **Multimodal A2A file-part encoding** NEW 73

`@ag-ui/a2a@0.0.7` supports multimodal content: image, audio, video, and document parts are now sent as A2A file parts, with inline data encoded as bytes and URLs as URIs, rather than as raw data parts.

— Explains encoding mechanism but no usage snippet. release/2026-10-13

10 **ag_ui_strands resume hooks and per-thread isolation** BREAKING 71

`ag_ui_strands` adds resume hook preservation and deduplication of backend results on native resume. In `ag_ui_strands@1.0.1`, conversation managers and template managers/summarizer factories are now isolated per thread and reject reuse; setups that previously shared these across threads or runs must be re-verified.

— Describes mechanism and migration note but no exact API name. release/2026-10-16 release/2026-10-12

11 **Mastra 2.0.0 peer dependencies and wire format changes** BREAKING 71

`@ag-ui/mastra@2.0.0` now requires `@ag-ui/core` and `@ag-ui/client` >=1.0.0, with the optional `@copilotkit/runtime` peer raised to ^1.76.0. It removes the legacy binary content part and the ignored `registerCopilotKit` serviceAdapter option, and tool-result URLs now serialize as URL items instead of base64 media — re-verify wire output.

— Names exact version requirements and removed options. release/2026-10-12

12 **a2uiGenerationToolNames field in A2UI middleware** NEW 70

Adds `a2uiGenerationToolNames` to `A2UIMiddlewareConfig` in `@ag-ui/a2ui-middleware`, letting apps opt in to disambiguating nested generation wrappers from ordinary tools while preserving existing wrapper defaults.

— Names the exact config field but gives no usage example. release/2026-10-19

13 **RUN_STARTED protocol version declaration across adapters**

IMPROVED 68

Watsonx adapters add a `RUN_STARTED` protocol version declaration (`protocolVersion` / `protocol_version`) as part of AG-UI 1.0 compatibility, and `AGUI.Server@1.0.1` 's run-start helper now sets `ProtocolVersion` to `AGUIProtocol.Version` on the `RUN_STARTED` event so the `protocolVersion` field appears in all event streams.

— Names exact fields but no code sample. `release/2026-10-16` `release/2026-10-12`

---

14 **Mastra getCapabilities accuracy changes** BREAKING 68

`getCapabilities` in `@ag-ui/mastra@2.0.0` now declares only guaranteed capabilities and warns when agent memory cannot be read; it no longer claims reasoning support or encryption, and declares shared state only for local agents with working memory.

— Describes exact behavior change, no code sample. `release/2026-10-12`

---

15 **adk 0.0.3 wire format and dependency changes** BREAKING 65

`@ag-ui/adk@0.0.3` now requires `@ag-ui/core` and `@ag-ui/client` >=1.0.0 as peer dependencies. Part renames and tool result content flattening change its wire output, and unsupported file sources are now dropped instead of forwarded.

— Names peer deps and behavior change but not the renamed fields. `release/2026-10-12`

---

16 **Managed agent session recovery across Claude SDK packages** IMPROVED 60

`@ag-ui/claude-managed-agents` , `ag-ui-claude-managed-agents` , and `AGUI.ClaudeManagedAgents` (TypeScript, Python, .NET) now recover managed agent sessions and pending approvals after interruption, and maintain that recovery state across stream handoff and cancellation.

— Behavior described but no flag or API named to invoke it. `release/2026-10-16`

THINNER COVERAGE BELOW

---

17 **Durable Object example Worker for Cloudflare agents** NEW 45

`@ag-ui/cloudflare-agents@0.0.2` adds a Durable Object example Worker, used by the AG-UI Dojo.

— Bare example mention, no setup details given. `release/2026-10-12`

└──▷ BREAKING ON UPGRADE

! Minimum toolkit dependency raised to `ag-ui-a2ui-toolkit 0.0.5` ; verify your lockfile.

! A2UI SDK pinned to `0.2.2` ; re-verify if you depend on another version.

! In `@ag-ui/claude-agent-sdk@0.0.7`, tool result serialization and frontend payload format changed; re-verify handling of tool names, result envelopes, and failure statuses.

! In `ag-ui-claude-sdk@0.2.2`, tool error reporting now uses the Python-native error flag; re-verify consumers that inspect tool result failure status or error semantics.

! In `ag-ui-claude-sdk@0.2.2`, structured tool result envelope and frontend payload serialization changed; re-check code parsing tool results or snapshots.

! `@ag-ui/mastra@2.0.0` requires `@ag-ui/core` and `@ag-ui/client` >=1.0.0; optional `@copilotkit/runtime` peer raised to ^1.76.0.

! `@ag-ui/mastra@2.0.0` removes the legacy CUSTOM `on_interrupt` event, `forwardedProps.command` resume input, and `emitInterruptOutcome` option; use the `RUN_FINISHED` interrupt outcome and `RunAgentInput.resume` instead.

! `@ag-ui/mastra@2.0.0` removes the legacy binary content part and the ignored `registerCopilotKit` serviceAdapter option.

! `@ag-ui/mastra@2.0.0`: tool-result URLs now serialize as URL items, not base64 media; re-verify wire output.

! `@ag-ui/mastra@2.0.0`: `getCapabilities` no longer claims reasoning support or encryption, and declares shared state only for local agents with working memory.

! `@ag-ui/mastra@2.0.0`: only one interrupt per run is emitted; multi-pause runs previously reported all pauses.

! `@ag-ui/mastra@2.0.0`: tool approval accepts only `true` / `{ approved: true }` to approve and declines only on cancel or `{ approved: false }`; other answers fail the run.

! `ag_ui_strands@1.0.1`: conversation managers and template managers/summarizer factories are now isolated per thread and reject reuse; re-verify setups that shared these across threads or runs.

! `@ag-ui/adk@0.0.3` requires `@ag-ui/core` and `@ag-ui/client` >=1.0.0 as peer dependencies.

! `@ag-ui/adk@0.0.3`: part renames and tool result content flattening change wire output; re-verify against providers.

! `@ag-ui/adk@0.0.3`: unsupported file sources are now dropped instead of forwarded.

! `ag-ui-langroid@0.2.0`: CORS credentials are no longer enabled for the default wildcard origin; pass explicit origins via the `origins` argument if browsers call the agent with cookies or credentials from another origin.

WAS THIS USEFUL? 👍 👎

sympozium

Feedback ⓘ v0.11.6 NOTES ↗ RANK

The Coordination Layer for Multi-Agent AI

> Sympozium's biggest changes this window are breaking: SympoziumPolicy toolGating is now enforced and Agents are locked to one mediated key each, alongside new gate-driven retry, celln artifact authority, bundled PostgreSQL, and CLI update subcommands.

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

01 SympoziumPolicy toolGating enforcement

BREAKING — 90

`SympoziumPolicy` `toolGating` is now enforced, so agents on a policy lose the tools it denies; the built-in restrictive policy denies `execute_command`, and policies using `ask` now fail validation — replace `ask` with `allow` or `deny`.

Apply a SympoziumPolicy that explicitly denies a tool, now that toolGating is enforced and takes effect immediately.

```
$ kubectl apply -f yaml/sympoziumpolicy-example.yaml
```

— Named policy field, concrete behaviour change, and runnable example v0.11.0

02 Per-Agent API key scoping and enforcement

BREAKING — 75

Agents now get per-Agent API keys scoped to the journey, with refusals and cells attributed to individual Agents. This is enforced as one key per Agent, mediated by default across any number of nodes via `spec.authRefs`; two Agents can no longer share a key Secret, and an AgentRun can no longer name a Secret that its Agent does not grant in `spec.authRefs`.

— Mechanism and breaking change named, but no example or command v0.11.0 v0.11.5

03 `sympozium update`/`upgrade` CLI subcommands

NEW — 70

Adds `sympozium update` and `sympozium upgrade` CLI subcommands for updating the tool in place, replacing manual brew/curl reinstall.

Pull the latest sympozium release without leaving the terminal — replaces the manual brew/curl reinstall loop.

```
$ sympozium update
```

— Exact command names and a runnable example provided v0.10.88

THINNER COVERAGE BELOW

04 **Bundled PostgreSQL in Helm chart** NEW 40

Bundles PostgreSQL in the Helm chart for mediated model access.

— Names Helm chart change but no config detail or example v0.11.0

05 **Gate-driven retry for response gates** NEW 35

Response gates can now trigger automatic retries of agent responses, enabling gate-driven retry.

— Describes what it does but no mechanism or usage detail v0.11.6

06 **Scoped artifact authority in celln backend** NEW 35

Adds scoped artifact authority and installed development proof to the `celln` backend.

— Names the backend but no mechanism or usage path v0.11.0

╰──▷ BREAKING ON UPGRADE

! Two Agents can no longer share a key Secret, and an AgentRun can no longer name a Secret that its Agent does not grant in `spec.authRefs`.

! `SympoziumPolicy` `toolGating` is now enforced — agents on a policy lose the tools it denies, and the built-in restrictive policy denies `execute_command`. Policies using `ask` fail validation; replace `ask` with `allow` or `deny`.

WAS THIS USEFUL? 👍 👎

skardi

Feedback ⓘ v0.7.0 NOTES ↗ RANK

Skardi is a self-improving context framework for ai agents

> Skardi's v0.6.0 and v0.7.0 releases add nine new data source packs (Notion, Feishu, RSS, Gmail, Discord, Outlook, OneDrive, Google Drive, Dropbox), an MCP-over-stdio CLI subcommand, a query audit ledger with ai_context tracking, named CLI contexts and console-brokered OAuth login, plus Lance upsert mode, a full-text predicate, S3-compatible Lance storage, and a native Windows CLI build.

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

01 **Query audit ledger with ai_context tracking** NEW 95

The server gains a `--query-audit-db` flag that records every query along with its `ai_context` (`purpose` , `session_id` , `task`) into a durable SQLite ledger indexed on (session_id, created_at); the `SKARDI_QUERY_AUDIT_PG_DSN` environment variable swaps that ledger storage to Postgres under the same fail-closed contract. Structured `ai_context` is now accepted as a body field on `POST /query` and via new `--purpose` and `--session-id` flags on `skardi query` , letting callers declare intent alongside SQL from either the API or CLI, with operator-controlled raw-SQL logging shipping in the same change.

Start a server with the query-audit ledger enabled so every agent query — and its declared intent — is durably recorded for pattern mining.

```
$ cargo run --release --bin skardi-server -- --ctx ctx.yaml --query-audit-db ./audit.db --port 8080
```

Switch the audit ledger from SQLite to Postgres for multi-instance deployments while keeping the same fail-closed contract.

```
$ SKARDI_QUERY_AUDIT_PG_DSN=postgres://user:pass@db:5432/audit cargo run --release --bin skardi-server -- --ctx ctx.yaml --port 8080
```

— Flags, env var, endpoint and runnable commands all named v0.6.0

02 **Nine new data source packs** NEW 85

skardi adds nine data source packs: Notion (milestone 5.3) with a `page_markdown` table rendering a Notion page as a single Markdown row; Feishu (milestone 5.4) with six tables, digit-string epoch handling, and boolean-pushdown engine support; RSS with

OPML config, HTML-to-Markdown sanitation, a TTL cache, a freshness engine, partition-per-feed execution, and an `AllowAll` default egress seam; Gmail (milestone 5.5); Discord (milestone 5.5) with keyset pagination engine support; Outlook (milestone 5.7) with `messages` and `mail_folders` tables over `nextLink` cursors; OneDrive (milestone 5.8) with `drive_items` and `drive_item_search` tables; Google Drive (milestone 5.9) with `files`, `drives`, and `file_permissions` tables; and Dropbox (milestone 5.10) with split-action cursor continuation.

— Every pack's named tables and milestones enumerated `v0.6.0`

03 **Named context management for CLI config** NEW 80

`~/.skardi/config.yaml` now supports named contexts (`current-context`, `contexts[].name/server/mode/workspace/token`), managed via the new `skardi config` subcommand, with a global `--context <NAME>` flag and `$SKARDI_CONTEXT` environment variable to select a context per command.

— Config schema, subcommand, flag and env var all named `v0.6.0`

04 **fts_match() full-text predicate for Lance** NEW 78

fts_match(column, query) is a new full-text predicate that can be pushed down into Lance queries, letting SQL filters such as WHERE fts_match(body, 'threat actor') execute at the storage layer.

Filter a Lance dataset by a full-text search term at query time, pushing the predicate down to the storage layer.

```
$ curl -X POST localhost:8080/query -H 'Content-Type:
application/json' -d '{"sql": "SELECT * FROM docs WHERE
fts_match(body, \'threat actor\') LIMIT 20", "ai_context":
{"purpose": "threat intel search", "session_id": "sess-42"}}'
```

— Named function with runnable query example `v0.7.0`

05 **Windows CLI build** NEW 78

A Windows (x86_64) build of the skardi CLI ships, installable via `irm https://raw.githubusercontent.com/SkardiLabs/skardi/main/install.ps1 | iex` from PowerShell, with its config file at `%USERPROFILE%\.skardi\config.yaml`.

Install the skardi CLI on a Windows analyst workstation so it can reach a running skardi-server.

```
$ irm
https://raw.githubusercontent.com/SkardiLabs/skardi/main/install
.ps1 | iex
```

— Exact install command and config path given v0.7.0

06 **Lance upsert mode with merge_keys** NEW 75

Lance destinations support `mode: upsert`, merging records on `merge_keys` to support incremental dataset updates instead of appending duplicates.

Push incremental embeddings into a Lance dataset without duplicating records — merge on a natural key instead of appending.

```yaml
# In your job/pipeline config, set mode and merge_keys on the
Lance destination:
kind: pipeline
metadata: { name: embed-sync }
spec:
  destination:
    type: lance
    mode: upsert
    merge_keys: [id]
```

— Config key and runnable example given, mechanism brief v0.7.0

07 **Pipeline execution audit ledger and /jobs/* auth requirement** 73

BREAKING

A single audit ledger now records both ad-hoc and pipeline calls with session attribution and a `run_id` bridge to the jobs ledger. Every `/jobs/*` endpoint now requires an authenticated session; unauthenticated callers that previously reached these endpoints will receive auth errors.

— Breaking auth change clearly scoped, no migration example v0.6.0

08 **Console-brokered OAuth login** NEW 65

`skardi login` and `skardi logout` subcommands authenticate via console-brokered OAuth that requires no OAuth client ID and works over SSH, including a workspace selector; they default to the hosted console.

— Subcommands and auth mechanism described, no example v0.6.0

09 **Apache AGE graph query support** NEW 65

Read-only Apache AGE graph support ships via `cypher_query` and `graph_schema` functions (milestone 1), with YAML catalog views and degraded registration (milestone 4).

— Functions named but no query example provided v0.6.0

10 **MCP-over-stdio CLI subcommand** NEW 62

`skardi mcp` is a new MCP-over-stdio binding that exposes all registered pipelines as MCP tools for local agent hosts such as Claude Desktop.

– Subcommand named, no example of invocation shown v0.6.0

THINNER COVERAGE BELOW

11 **CLI telemetry identification and opt-out** NEW 50

The CLI identifies itself in its User-Agent header and honours the `DO_NOT_TRACK` environment variable to disable telemetry.

– Named env var but no example of use v0.7.0

12 **S3-compatible Lance storage** NEW 40

Lance datasets can now be read from and written to S3-compatible object stores.

– States capability but no config keys or example given v0.7.0

13 **Native Markdown and plain-text ingestion** NEW 40

skardi reads Markdown and plain text documents natively, without routing them through liteparse.

– Mechanism noted but no usage detail or example v0.7.0

14 **Cloud gateway `run` and `pipeline` subcommands** NEW 40

`skardi run` and `skardi pipeline` subcommands execute against a cloud gateway.

– Subcommands named but no mechanism or example v0.6.0

15 **`chunk_parts` and `json_pack` runtime UDFs** NEW 35

`chunk_parts` and `json_pack` runtime UDFs are now available to generated ETL bundles.

– Function names given, no usage or mechanism detail v0.6.0

╰──▷ BREAKING ON UPGRADE

! Every `/jobs/*` endpoint now requires an authenticated session — unauthenticated callers that previously hit these endpoints will receive auth errors.

WAS THIS USEFUL? 👍 👎

Anthropic Claude Agent SDK Python

Feedback

v0.2.166 NOTES ↗ RANK

The Claude Agent SDK for Python is a developer library for building Python applications that run Claude-powered agents with tools, file access, and autonomous multi-step task execution.

> Claude Agent SDK v0.2.166 strengthens its Python type surface with a fully typed init payload and hook outputs, while also tightening the sandbox violation-reporting type in a breaking way.

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

01 **Typed init payload and hook outputs** NEW 80

Adds a `SystemInitData` TypedDict (with nested types `SystemInitMcpServer`, `SystemInitPlugin`, `SystemInitPluginError`, and `FastModeState` / `FastModeDisabledReason` literals) to fully type the init system message payload, accessible via cast(SystemInitData, msg.data) with fields including `capabilities`, `model`, and `mcp_servers`. Also adds typed output classes and previously missing input fields for hooks, replacing untyped dicts with precise type information for hook implementations.

— Names exact types/fields and the cast usage pattern. v0.2.166

02 **SandboxSettings.ignoreViolations type change** BREAKING 73

`SandboxSettings.ignoreViolations` type changed from a `TypedDict` with `file` / `network` keys to `dict[str, list[str]]` (a map from command-pattern substring to violation-text substrings). Code relying on the old `file` / `network` keys, or strict type checkers that accepted the old shape, must be updated to the new `{'*': [...]}` form.

— Gives exact before/after type and migration shape. v0.2.166

└──▷ BREAKING ON UPGRADE

! `SandboxSettings.ignoreViolations` type changed from a `TypedDict` with `file` / `network` keys to `dict[str, list[str]]` (a map from command-pattern substring to violation-text substrings); code that relied on the old `file` / `network` keys or that strict type checkers accepted under the old shape will need to be updated to the correct `{'*': [...]}` form.

WAS THIS USEFUL? 👍 👎

octo-cli

Feedback ⓘ v0.18.0 NOTES ↗ RANK

Metadata-driven CLI for AI Agent Bots — 48 operations across 7 domains, structured JSON envelope I/O, zero interactive prompts.

> octo-cli v0.18.0 adds version-aware HTML reading for guarded publishing and expands the docs domain with conditional formatting, safe sheet cleaning, and worksheet discovery.

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

01 **Guarded HTML publishing via source+version reads** NEW

45

The `html` domain can now read HTML document source together with its version, enabling guarded (version-checked) publishing flows.

— Explains mechanism but no concrete command names or examples given. v0.18.0

02 **Docs domain: formatting, cleaning, and worksheet discovery** NEW

35

The `docs` CLI gains conditional formatting and safe sheet cleaning commands, plus discovery and scoped reading of individual worksheets.

— Names the domain and capabilities but no specific commands or flags. v0.18.0

WAS THIS USEFUL? 👍 👎

guild

Feedback ⓘ v0.4.0 NOTES ↗ RANK

Shared context, memory, and task coordination across AI coding agents. Single Go binary, local SQLite, hybrid keyword and semantic search.

> Guild 0.4.0 introduces a resident daemon with shared sessions and file watching, reworks hook installation into dedicated subcommands, and adds Homebrew packaging, broader platform support, retrieval improvements, and signed release verification — alongside a breaking quest-search migration.

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

01 Resident daemon with shared sessions and file watching NEW

88

Guild now runs as a resident daemon providing shared sessions across agents, lease heartbeats, and automatic recovery, plus project file and git watching that generates staleness signals and renewal quests. The `[daemon] watch = false` config key in `~/.guild/config` disables file watching and falls back to query-time staleness checks when file descriptor limits are hit.

Suppress file-watcher descriptor exhaustion on macOS with large project registries by disabling watching at the daemon level.

ini

```
[daemon]
watch = false
```

— Named config key and file path with clear fallback behavior. v0.4.0

02 Hooks lifecycle management subcommands NEW

83

Adds `guild hooks install` to wire Guild into harness lifecycle hooks, along with `guild hooks sync`, `guild hooks list`, `guild hooks diff`, and `guild hooks scan` for managing those installations.

Wire Guild into all detected harness lifecycle hooks so context is injected proactively at session start and before compaction.

```
$ guild hooks install
```

After editing the shared hooks config, propagate changes to all harnesses and verify sync status.

```
$ guild hooks sync && guild hooks list
```

— All five subcommands named with runnable examples. v0.4.0

03 **Homebrew 7 cask DSL support** NEW 70

Supports the Homebrew 7 cask DSL, enabling installation and upgrade via `brew install --cask mathomhaus/tap/guild` and `brew upgrade --cask mathomhaus/tap/guild`.

— Exact install and upgrade commands provided. v0.4.0

04 **Migration 014 rebuilds quest search data** BREAKING 70

Migration 014 rebuilds quest search data and discards old quest vectors on first startup; rolling back requires restoring the matching pre-upgrade database backup. Users must stop old Guild MCP processes and daemons and back up both SQLite databases and their sidecar files before upgrading.

— Clear migration steps but no exact file paths given. v0.4.0

THINNER COVERAGE BELOW

05 **MCP setup detects existing Codex registrations** IMPROVED 50

MCP setup now detects existing Codex registrations, preserves existing HTTP registrations unless replacement is explicitly requested, and reports clients whose CLI is unavailable.

— Describes behavior but no command or flag given. v0.4.0

06 **Retrieval improvements: isolation, partial embeddings, fallback** IMPROVED 50

Improves retrieval with project isolation, support for partially embedded corpora, fresh-vector eligibility checks, exact-title lookup, and keyword fallback.

— Lists mechanisms but no command or config to invoke. v0.4.0

07 **Release asset verification with SHA256 and Sigstore** NEW 45

Adds release asset verification with SHA256 checksums and a Sigstore signature and certificate for independent verification.

— Names verification methods but no command shown. v0.4.0

08 **Module-based CLI and MCP surface architecture** NEW 30

Introduces module-based CLI and MCP surfaces with explicit lifecycle-hook installation and optional evaluation, compression, and observability capabilities.

— No named modules, flags, or endpoints given. v0.4.0

09 **Native runtime support for Intel Mac and Windows ARM64** NEW

30

Adds native runtime coverage for Intel Mac and Windows x64/ARM64.

— Bare platform list with no setup detail. v0.4.0

└──▷ **BREAKING ON UPGRADE**

! Migration 014 rebuilds quest search data and discards old quest vectors on first startup; rolling back requires restoring the matching pre-upgrade database backup. Stop old Guild MCP processes and daemons and back up both SQLite databases and their sidecar files before upgrading.

WAS THIS USEFUL? 👍 👎

tinybot

Feedback ⓘ v1.4.0 NOTES ↗ RANK

Tinybot is a lightweight personal AI Agent that is constantly evolving

> Tinybot v1.4.0 brings multi-member Teams collaboration into the Chat view and side panel, with dynamic recruitment and per-member tool permissions, alongside rendering performance work for long conversations.

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

01 **Teams collaboration in Chat and side panel** NEW 70

Integrates Teams collaboration into the Chat view and side panel, letting member activity, task progress, history, output files, and usage be viewed together with pause, cancel, and retry controls. Supports dynamically recruiting members mid-conversation, configuring tool permissions per member, and executing tasks with isolated sessions and result handoff. Adds a member-switcher tab and task-plan display, supporting viewing of different execution attempts per member, expanding reasoning and full tool output, and preserving scroll position and expansion state.

– Describes mechanism and UI scope but no exact config keys or commands v1.4.0

THINNER COVERAGE BELOW

02 **Rendering performance for long conversations** IMPROVED 50

Optimizes streaming rendering for long replies, eliminating per-character animation nodes while fully preserving Markdown content and code-block action states. Also reduces redundant rendering of collapsed details and unchanged tool rows, lowering overhead during sustained output and scrolling in long conversations.

– Explains mechanism but no measurable benchmarks or config options v1.4.0

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Inference Providers & GPU Clouds

AWS Amazon Bedrock

Feedback ⓘ changelog · Oct 25, 2026

CHANGELOG ↗ RANK

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

> Amazon Bedrock shipped agent-level retrieval instructions for knowledge bases, prompt shortening in Advanced Prompt Optimization, a new Standard tier for Guardrails contextual grounding checks, and structured outputs support extended to Anthropic Claude models.

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

01 **Retrieval instructions for agentic knowledge base retrieval** NEW 72

Adds `retrievalInstructionConfiguration` to agentic retrieval on fully managed Amazon Bedrock knowledge bases, letting you steer how the agent searches knowledge bases and answers. Amazon Bedrock reviews each instruction before it takes effect, and pinned immutable versions are supported.

– Names exact config field and behaviour, but no sample payload given

changelog-20261025-58cf0215

02 **Prompt shortening in Advanced Prompt Optimization** IMPROVED 65

Advanced Prompt Optimization now supports prompt shortening per prompt template, reducing prompt length while maintaining or improving evaluation scores. Results now include median time to first token and average and median latency, and application inference profiles are accepted as both target models and judge models.

– Describes mechanism and new metrics but no exact API call changelog-20261025-58cf0215

THINNER COVERAGE BELOW

03 **Standard tier for Guardrails contextual grounding checks** NEW 45

Amazon Bedrock Guardrails contextual grounding checks gain a Standard tier, supporting responses that include reasoning or conversational content, more consistent scores, and additional languages.

– Names the tier and benefits but no config detail changelog-20261025-58cf0215

04 **Structured outputs for Claude models** IMPROVED 30

Structured outputs support is extended to Anthropic Claude models on both endpoints.

— Thin one-line description with no mechanism or endpoint detail

changelog-20261025-58cf0215

WAS THIS USEFUL? 👍 👎

Cloudflare Workers AI

Feedback ⓘ changelog · Oct 25, 2026

CHANGELOG ↗ RANK

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

> This window's updates center on Workers Builds gaining npm/Yarn/pnpm/Bun version pinning and Flags adding nested-context targeting, alongside a cluster of Cloudflare One (Zero Trust) Windows/macOS client changes, a breaking 7-day restriction on the D1 undelete API, and a redesigned Cloudflare Radar homepage.

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

01 Nested context support in Flags evaluation IMPROVED

90

Flags feature-flag evaluation context now supports nested objects, arrays, and `null` across the Workers binding (`env.FLAGS.getBooleanValue`), the evaluation REST API, and the TypeScript, Python, and Go OpenFeature SDKs, enabling dot-separated path targeting (e.g. `profile.plan`) and `has` / `not_has` operators for array membership.

Target enterprise beta users with a feature flag by passing nested profile and tags in the Workers binding evaluation context.

```javascript
const enabled = await env.FLAGS.getBooleanValue('new-checkout',
false, {
  targetingKey: 'user-42',
  profile: { plan: 'enterprise', id: 'u42' },
  tags: ['beta', 'early-access']
});
```

— Binding method, SDKs, and operators named with a runnable example

changelog-20261025-592c4d71

02 **MDM client configuration parameters for Windows** NEW

90

Cloudflare One Client for Windows adds `mdm_file_debounce_secs` (1–60), `mdm_file_unavailable_debounce_secs` (0–300), `managed_network_probe_interval_secs` (30–300, default 30), and `managed_network_probe_count` (0 disables, 1–10, or -1 for continuous; default 1), controlling how long the client waits before applying changed or missing MDM files and how it rechecks for a managed network after registration or network changes.

— Four config keys with exact ranges and defaults named `changelog-20261025-592c4d71`

03 **npm/Yarn/pnpm/Bun version pinning in Workers Builds** NEW

85

Workers Builds now reads a `packageManager` field in `package.json` (e.g. `npm@11.19.1`) to pin the exact npm, Yarn, pnpm, or Bun version used before dependency installation, so CI installs match local lockfiles.

Pin an exact npm version in Workers Builds so lockfiles written locally install identically in CI.

```json
{"packageManager": "npm@11.19.1"}
```

— Exact config file, field, and example value given `changelog-20261025-592c4d71`

04 **Traceroute recording in warp-diag** NEW

60

When a Zero Trust tunnel degrades or fails to connect, the client now records a traceroute of the network path outside the tunnel, included in `warp-diag` output.

— Named diagnostic output `warp-diag` gives a concrete starting point

`changelog-20261025-592c4d71`

05 **D1 undelete API restricted to 7-day window** BREAKING

60

The D1 undelete API now only accepts requests for databases deleted within the last 7 days; requests for databases deleted earlier than that return an error, so previously-successful undelete calls may now fail.

— Exact time window named but no endpoint path given `changelog-20261025-592c4d71`

THINNER COVERAGE BELOW

06 **macOS 14 support dropped for Cloudflare One Client** BREAKING

55

Cloudflare One Client version 2026.9.1555.1 drops support for macOS 14; macOS 15.0 is now the minimum supported version.

— Exact version numbers given for migration planning changelog-20261025-592c4d71

07 **Automatic Global Acceleration endpoint selection** NEW

50

Cloudflare One Client for Windows and macOS now automatically connects to Global Acceleration endpoints when in-region and falls back to standard anycast endpoints otherwise, reconnecting automatically on network changes.

— Describes behavior but no config surface to act on changelog-20261025-592c4d71

08 **Cloudflare Radar homepage redesign** IMPROVED

45

The Cloudflare Radar homepage has been redesigned with an interactive global traffic/outage map, curated tabbed insights, and a rebuilt UI using Kumo components.

— Names UI elements but gives no navigation path or API changelog-20261025-592c4d71

09 **Tamper-protection for client uninstall** NEW

35

Cloudflare One Client for Windows adds a tamper-protection feature requiring an administrator-generated code to uninstall the client.

— Single-sentence description, no further mechanism given changelog-20261025-592c4d71

10 **MTU degradation banner** NEW

30

Cloudflare One Client for Windows and macOS now shows an MTU degradation banner when the network limits the tunnel MTU to a degraded or critically low value.

— Describes trigger but no further mechanism or UI path changelog-20261025-592c4d71

11 **Post-quantum encryption status indicator** NEW

20

The consumer client home screen now shows a post-quantum encryption status indicator.

— Bare description, no mechanism or additional detail changelog-20261025-592c4d71

└──▷ BREAKING ON UPGRADE

! The D1 undelete API now rejects requests for databases deleted more than 7 days ago, returning an error — undelete requests that previously would have succeeded may now fail.

! macOS 14 is no longer supported by Cloudflare One Client version 2026.9.1555.1; macOS 15.0 is the new minimum supported version.

WAS THIS USEFUL? 👍 👎

Eigen Labs Darkbloom

Feedback ⓘ v0.10.1 NOTES ↗ CODE ↗

RANK

Darkbloom is a for the innovators pushing the frontier.

> Darkbloom v0.10.1 introduces a native macOS desktop app and raises the minimum macOS version required for new installations.

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

01 **macOS 27 required for new installs** BREAKING 60

New provider installations now require macOS 27 or later; running the installer on macOS 26 or earlier exits before downloading anything. Existing installations are unaffected and can still reinstall or repair, with an upgrade notice shown.

– Clear before/after and remediation path for existing users v0.10.1

THINNER COVERAGE BELOW

02 **macOS desktop app backed by Swift CLI** NEW 55

Darkbloom ships a new macOS desktop app built with Electron and backed by the existing Swift CLI. Desktop builds are signed, notarized, and published as GitHub Releases tagged `desktop-vX.Y.Z`.

– Names release tag format but no install command given v0.10.1

└──▷ **BREAKING ON UPGRADE**

! New provider installations now require macOS 27 or later; running the installer on macOS 26 or earlier exits before downloading anything (existing installations can still reinstall or repair with an upgrade notice).

WAS THIS USEFUL? 👍 👎

CoreWeave

Feedback ⓘ announced · Oct 22, 2026 BLOG ↗

CoreWeave is an AI cloud platform offering GPU compute, managed Kubernetes and storage for training and inference.

NVIDIA Vera Rubin NVL72 is now in limited availability on CoreWeave Cloud for production training and inference via CKS, CoreWeave Inference, SUNK, AI Object Storage, and Mission Control.

NEW SUPPORTED PLATFORM NVIDIA Vera Rubin NVL72 on CoreWeave Cloud

WAS THIS USEFUL? 👍 👎

Modal

Feedback ⓘ announced · Oct 22, 2026 BLOG ↗ RANK

Modal is a cloud infrastructure for teams that develop, train, and serve AI applications at scale.

> Modal launched Spindle, an open-source Tinker-compatible training server that deploys into a Modal workspace with a single command and supports customizable configs and multi-tenant LoRA training.

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

01 Custom Spindle configs via subclassing NEW

83

Spindle configurations can be extended by subclassing built-in config classes and overriding fields such as `max_context_length` and `context_parallel_size` to customize model and training parameters, as shown extending a Qwen config to raise context length to `256k` and set `context_parallel_size = 4`.

Extend a built-in Spindle model config to bump context length to 256k and enable context parallelism for long-horizon tasks, without forking the server.

python

```python
class Config(Qwen35_9B_16k):
    max_context_length = "256k"  # was "16k"
    context_parallel_size = 4    # was 1
```

— Concrete fields and a runnable code example given launch-20261022-4d3992c9

02 Spindle training server deployment NEW

75

Spindle, an open-source Tinker-compatible post-training server, deploys into a Modal workspace with a single `spindle deploy` command, enabling full-parameter training and multi-LoRA on open models without relying on a closed-source training API.

Deploy a Spindle training server into your Modal workspace so your existing Tinker-compatible training scripts can point at it instead of a closed-source API.

```
$ spindle deploy
```

— Names exact deploy command and capability, but little mechanism detail

launch-20261022-4d3992c9

THINNER COVERAGE BELOW

03 Multi-LoRA multi-tenant training NEW

45

Spindle supports multi-LoRA multi-tenant training, letting hundreds of concurrent training runs share the same trainer and inference GPUs, which reduces per-run cost significantly compared to closed-source APIs.

— Explains mechanism and benefit but no concrete numbers or interface

launch-20261022-4d3992c9

WAS THIS USEFUL? 👍 👎

◆ Agentic Infrastructure

Hugging Face

Feedback ⓘ

External Collaborators from Repo Settings | NOTES ↗ | RANK

Hugging Face is an open platform for building, sharing, and deploying machine-learning models, datasets, and applications.

> Hugging Face now lets organization admins invite external collaborators directly from private repository settings, with access scoped to a single Resource Group.

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

01 **External collaborators from repo settings** NEW 85

Org admins can invite external collaborators directly from a private repository's Settings > Collaborators page, granting read or write access scoped only to that repository's Resource Group rather than the whole organization. A new External invitations toggle under Organization Settings > Resource Groups > [resource group] > Settings lets admins allow resource group admins to also invite external collaborators; it is off by default and available on Team and Enterprise plans.

Invite an external collaborator to a private repo with scoped access, without adding them to the full organization.

> 📍 In the Hugging Face UI, go to your private repository › Settings › Collaborators, then use Invite external collaborator to grant read or write access to that repository's Resource Group.

Allow resource group admins to invite external collaborators — useful for delegating access management on Enterprise or Team plans.

> 📍 In the Hugging Face UI, go to Organization Settings › Resource Groups › [your resource group] › Settings, then enable External invitations.

— Names exact UI navigation, scope, plan availability, and default state.

External Collaborators from Repo Settin…

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.8.197 NOTES ↗ RANK

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

> HyperFrames shipped a fuller motion-blur rendering pipeline with automatic tier selection and diagnostics, a breaking switch to arm64-by-default Lambda renders with Docker-free ECR push, and a string of Studio and CLI refinements spanning preview scrubbing, blur-triggered events, local GSAP serving, and recording parsing.

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

01 Lambda renders default to arm64 with Docker-free ECR push

88

BREAKING

Lambda cloud renders now default to arm64; pass `--video-arch x86_64` to keep the previous architecture. Enables pushing a prebuilt image to ECR without Docker on AWS Lambda deploys, runs Lambda render chunks on an optional arm64 image function to reduce per-render cost, and attaches and pins an arm64 video image to Lambda releases.

— Exact override flag and migration note given, directly actionable. v0.8.188

02 Studio motion blur panel: rendering, events, and diagnostics

NEW

78

Adds a motion blur panel to Studio with three levels: Off, Subtle, and Strong, turns on events with blur to enable blur-triggered interactions, introduces automatic motion-blur tier selection with defaults of `240°` shutter angle and `32` samples replacing manual opt-out, falls back to per-element filters when capture cannot run, wires capture motion blur into renders and Studio while automatically disabling it with an explanation on pages where it cannot safely apply, reports why blur is off and what each render session did via diagnostics per timeline route, and tracks accelerated canvases before any page script runs for earlier blur-eligibility detection.

— Full mechanism and defaults given, but control is UI-only, no CLI flag named. Names the UI panel but not the interaction mechanism.

v0.8.186 v0.8.190 v0.8.191 v0.8.194

03 Local GSAP dependency mirror in Studio NEW

60

Enables a local GSAP dependency mirror in Studio, allowing air-gapped or controlled environments to serve GSAP locally instead of from a remote CDN.

— Explains the mechanism and use case, but no config key named. v0.8.194

THINNER COVERAGE BELOW

04 **CLI check command progress reporting and overlap** IMPROVED 56

CLI `check` now writes a progress line to stderr per sample and overlaps batch and frame crafting.

— Named command and behaviour given, but no numbers on speed gain. v0.8.191

05 **Lint warning for hidden round-capped dash lines** NEW 55

Adds a Lint warning when a round-capped line hidden by its dash settings can still produce a visible dot.

— Names the lint rule and exact condition it catches. v0.8.197

06 **hyperframes history resilient listing** IMPROVED 54

Adds `hyperframes history` listing of entries even when the current run cannot write to the history file.

— Exact subcommand named with the fallback behaviour it covers. v0.8.186

07 **init --video framing to clip shape** IMPROVED 54

Extends `init --video` to frame the project to the clip's shape so nothing is cropped.

— Exact flag named with the concrete cropping fix it delivers. v0.8.186

08 **Preview seek bar in Studio** NEW 42

Adds a preview seek bar to Studio for scrubbing through video previews.

— Clear UI feature but no further mechanism described. v0.8.190

09 **CLI recording parsing into voice, effect, and music lines** NEW 35

Reads a whole recording as voice, effect, and music lines from the CLI.

— No command name or output format given, just a capability statement. v0.8.186

10 **Disable orphan browser cleanup in CLI** NEW 30

Adds the ability to disable orphan browser cleanup in the CLI.

— No flag name given, just the capability. v0.8.194

└──▷ **BREAKING ON UPGRADE**

! Lambda cloud renders now default to arm64; any pipeline that assumed x86_64 must pass `--video-arch x86_64` explicitly to preserve the previous architecture.

WAS THIS USEFUL? 👍 👎

Tracer Cloud OpenSRE

Feedback ⓘ v0.1.2026.10.24 NOTES ↗

RANK

OpenSRE is an open-source toolkit for building AI SRE agents.

> OpenSRE expanded its LLM authentication options with a subscription adapter registry, gated Grok OAuth, and browser-based ChatGPT subscription login, while also adding live reply streaming, a coding agent fallback, and Simplified Chinese documentation.

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

01 Subscription-based LLM authentication adapters NEW

35

Adds a subscription adapter registry and gated Grok OAuth support for LLM integrations, plus browser-based ChatGPT subscription login via feat(auth) integration.

– Names adapters and OAuth but no config or usage detail v0.1.2026.10.23 v0.1.2026.10.24

02 Live AI reply streaming with auto-continue IMPROVED

25

Streams AI replies live and automatically continues responses that get cut off.

– Describes behaviour but no mechanism or config v0.1.2026.10.23

03 Built-in coding agent fallback NEW

25

Adds a built-in coding agent fallback for when the primary coding agent is unavailable.

– States purpose only, no detail on activation or scope v0.1.2026.10.23

04 Simplified Chinese README added NEW

20

Adds a Simplified Chinese README, extending localization coverage.

– Minimal documentation addition with no further detail v0.1.2026.10.23

WAS THIS USEFUL? 👍 👎

Anthropic TypeScript SDK

Feedback ⓘ sdk-v1.3.0

NOTES ↗ RANK

The Anthropic TypeScript SDK provides typed JavaScript and TypeScript clients for Claude APIs, streaming, and tool use.

> The Anthropic TypeScript SDK added Claude Tag endpoints for Enterprise organizations and new usage/cost reports by dimension, with older report endpoints marked legacy.

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

01 **Usage and cost reports by dimension** NEW 33

Adds usage and cost reports broken down by dimension, while marking the older report endpoints as legacy.

— Describes change but no endpoint names or dimensions listed sdk-v1.2.0

02 **Claude Tag endpoints for Enterprise orgs** NEW 30

Adds Claude Tag endpoints to the SDK, available for Claude Enterprise organizations.

— Names feature but no endpoint paths or methods given sdk-v1.3.0

WAS THIS USEFUL? 👍 👎

Anthropic Python SDK

Feedback ⓘ v2.3.0 NOTES ↗ RANK

The Anthropic Python SDK provides typed Python clients for Claude APIs, streaming responses, and tool use.

> Anthropic Python SDK added Claude Tag endpoints for Enterprise orgs, message threads via the API, and new usage/cost reports broken down by dimension.

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

01 **Claude Tag endpoints for Enterprise orgs** NEW 30

Adds Claude Tag endpoints for Claude Enterprise organizations.

— Names endpoint area but no methods, paths, or usage v2.3.0

02 **Usage and cost reports by dimension** NEW 30

Adds usage and cost reports broken down by dimension; older reports are now marked legacy.

— States behavior change but no field or API names v2.2.0

03 **Message threads via the API** NEW 20

Adds support for message threads via the API.

— Bare feature name, no endpoint or usage detail v2.2.0

WAS THIS USEFUL? 👍 👎

Hugging Face Diffusers

Feedback ⓘ v0.42.0 NOTES ↗

RANK

Diffusers provides pretrained diffusion pipelines, schedulers, and models for image, video, and audio generation in PyTorch.

> Diffusers v0.42.0 adds a wave of new generation pipelines — FLUX.3 Image Klein, Bria Fibo 2.0, Kandinsky 6, SANA-WM, RFD4-Proteína, and a 48kHz audio VAE — alongside TorchTPU backend support, LoKr adapter loading, and breaking changes to Python version support and latent handling in FLUX/Krea 2/LTX-Video.

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

01 FLUX.3 Image Klein via ModularPipeline NEW

90

Adds FLUX.3 Image Klein support via `ModularPipeline`, with single-file transformer loading, BFL FP8r and NVFP4 checkpoints, an integrated prompt upsampler, and layout instructions with bounding boxes.

Generate an image with FLUX.3 Image Klein via ModularPipeline, loading weights separately after pipeline construction.

python

```python
from diffusers import ModularPipeline

pipeline = ModularPipeline.from_pretrained('black-forest-labs/FLUX.3-Image-Klein')
pipeline.load_components()
output = pipeline(prompt='A photorealistic sunset over the ocean')
output.images[0].save('output.png')
```

— Named checkpoint formats plus a runnable loading example v0.42.0

02 Kandinsky 6 video pipelines and PiflowScheduler NEW

80

Adds `Kandinsky6TI2VAPipeline` for text-to-video and image-to-video generation with synchronized audio, `Kandinsky6SRPipeline` for video super-resolution, and a new `PiflowScheduler` for distilled model variants.

Generate video with synchronized audio from a text prompt using Kandinsky 6.

```python
from diffusers import Kandinsky6TI2VAPipeline
import torch

pipeline = Kandinsky6TI2VAPipeline.from_pretrained('ai-forever/Kandinsky-6', torch_dtype=torch.float16)
pipeline.to('cuda')
video = pipeline(prompt='A timelapse of storm clouds rolling over mountains').frames[0]
video[0].save('output.gif')
```

— Three named classes grouped, backed by a runnable example v0.42.0

---

03 **BriaFibo2Pipeline for structured image generation and editing** NEW 75

Adds `BriaFibo2Pipeline` for image generation from structured JSON captions and editing with up to five reference images, including a dev checkpoint and an `edit-turbo` checkpoint that runs in four steps without classifier-free guidance, plus DreamBooth LoRA training examples for Bria Fibo 2.0 text-to-image and image editing.

— Names checkpoints and mechanism but no runnable example given v0.42.0

---

04 **Unpacked latents required for FLUX, Krea 2, LTX-Video** BREAKING 75

FLUX, Krea 2, and LTX-Video now unpack latents in their core denoise groups — custom workflows must pass unpacked latents to decode blocks. The fallback for packed latents is deprecated for removal in 0.47.0.

— Clear migration requirement and deprecation deadline given v0.42.0

---

05 **SANA-WM camera-controlled image-to-video** NEW 62

Adds SANA-WM pipeline for camera-controlled image-to-video generation from a starting image, text prompt, and camera trajectory; camera control accepts explicit poses or a WASD/IJKL action string.

— Mechanism described but no code example provided v0.42.0

---

06 **motion_speed parameter for LTX-2** NEW 60

Adds `motion_speed` parameter to LTX-2 for controlling the positional time base independently of playback frame rate.

— Named parameter with clear purpose, no usage snippet v0.42.0

THINNER COVERAGE BELOW

07 **RFD4-Proteína biomolecular design pipeline** NEW 56

Adds RFD4-Proteína pipeline for biomolecular design, jointly generating protein sequences and all-atom structures while conditioning on binding targets, motifs, or ligands.

— Clear capability description, no usage example or API shown v0.42.0

08 **TorchTPU backend integration** NEW 56

Adds TorchTPU backend integration covering eager execution, `torch.compile`, and tensor parallelism; tensor parallelism also gains Intel XPU support.

— Names execution modes and XPU support, no setup steps v0.42.0

09 **Python and huggingface_hub version requirements raised** BREAKING 55

Python 3.10 is no longer supported; Python 3.11 or newer is required. The supported `huggingface_hub` version range is now `>=1.32.0,<3.0`.

— Exact version constraints named, straightforward to check v0.42.0

10 **LoKr adapter loading for Z-Image and FLUX.2** NEW 48

Adds LoKr (LyCORIS Kronecker-product) adapter loading for Z-Image and FLUX.2, including Klein.

— Named adapter type and models, no loading example v0.42.0

11 **AutoencoderKLKVAEAudio for 48kHz audio** NEW 32

Adds `AutoencoderKLKVAEAudio` for encoding and decoding 48 kHz audio.

— Bare name and sample rate, no usage shown v0.42.0

12 **Qwen-Image 2.1 modular pipeline and tensor parallelism** NEW 30

Adds modular pipeline and tensor parallelism support for Qwen-Image 2.1.

— One-line addition with no mechanism or example v0.42.0

╰──▷ BREAKING ON UPGRADE

! Python 3.10 is no longer supported; Python 3.11 or newer is required.

! FLUX, Krea 2, and LTX-Video now unpack latents in their core denoise groups — custom workflows must pass unpacked latents to decode blocks. The fallback for packed latents is deprecated for removal in 0.47.0.

! The supported `huggingface_hub` version range is now `>=1.32.0,<3.0`.

WAS THIS USEFUL? 👍 👎

feder-cr gitai

Feedback ⓘ v0.4.0 NOTES ↗ RANK

Git designed for AI agents.

> gitai's core UserLand RPC for agent-driven landing of changes went from a v0.1.0 debut to repeated throughput rewrites (up to ~2,500 lands/min with sub-2s p99 latency), while a vendor-neutral conflict resolver grew into a full multi-model proposal/verification workflow; the window also added storage checkpoints and GC, hardened branch-name resolution against agent attacks, and shipped Windows support and a `graft repo create` command.

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

01 UserLand RPC and landing throughput optimizations NEW

93

Introduces the `UserLand` RPC (v0.1.0) that rebases and lands changes with compare-and-swap semantics, batching concurrent lands on a single branch. v0.2.0 writes and durably flushes all objects as a single pack instead of fsyncing each object ~3 times, reduces the landing decision from 9 git processes to a single range listing, and unifies the single and batched land paths, reaching 1,862–2,544 lands/min at 50 agents (up from 240–272) with p99 latency dropping from 6.3–7.0s to 1.5–1.8s, and reads object format from repository config instead of spawning a second `git rev-parse` for `attr.tree`. v0.3.1 further improves the land path so file-bucket uploads fsync only directories not yet known to be durable (~1,234 → ~1,650 lands/min on WSL2 with SQLite and 10 writers) and adds the `graft_storage_preparing_step_seconds{step,kind}` metric (steps: record, walk, pack, upload) for authoritative writes. v0.3.0 reduces `UserCommitFiles` from 10–15 git processes to 6 (one reader, one tree writer, lazy trees, exact branch resolution), cutting p99 write latency from 600ms to 444ms at 50 agents, adds an authoritative-mode write path raising lands/min from 677–972 to 1,411–1,673 under 50–100 agents, adds changelog group commit so concurrent writes share one fsync in order, caps `graft.land.batch.max_size` at 100, and adds SDK `LandOptions.RetryUnavailable` (retries `Unavailable` up to 5 times with jittered backoff for merge and fast-forward lands, reported via `LandResult` retry counts).

— Numbers, config keys, and metric names given across every release. v0.3.1 v0.3.0 v0.2.0 v0.1.0

02 Conflict resolution workflow with multi-model agreement

IMPROVED

91

v0.1.0 introduced a vendor-neutral conflict resolver protocol in `UserLand`, letting any LLM provider (or none) resolve merge conflicts. v0.3.0 built this into a full workflow: every conflicting file is written as a conflict package with its three sides, hunks, and the

reason it was not resolved; the CLI re-lands the agent's resolution with `--resolved`; unresolved cases reach the agent as a proposal with a report of what the server checked; `agree_with` lets a second model from another vendor resolve the same file, holding it for the agent if the two differ or a composition check finds a dropped or duplicated change; `verify.command` gates automatic landing on the project's tests passing on the exact tree, otherwise every model resolution becomes a proposal; resolved files must declare `origin: "rule"` or `origin: "model"`, with unmarked answers treated as model resolutions; the lane is tuned via `[graft.review]` `max_open_reviews` (default 1000), `rules_only`, and `inline_timeout` (default 1s). `graft-resolve` and the server must be upgraded together since resolution capability (`proposal`, `hold`) is negotiated, and rerere entries written by v0.1/v0.2 are never replayed under the new store.

— Config keys, CLI flag, and upgrade constraints fully named. v0.3.0 v0.1.0

---

03 **Agent access control and branch-resolution hardening** BREAKING

86

v0.1.0 adds standalone mode for coding agents with per-agent tokens, an RPC allowlist, and branch policies, requiring no GitLab. v0.4.0 enforces exact branch-name resolution in every operations RPC (`UserFFBranch`, `UserMergeBranch`, `UserDeleteBranch`, `UserUpdateSubmodule`, `UserCherryPick`, `UserRevert`, `UserApplyPatch`, `UserRebaseConfirmable`, and default branch), preventing reference-aliasing attacks where a tag could stand in for a branch, and defines a strict resolution order for start points (existing branch beats same-named tag or `refs/x`; full object ID, `refs/...`, `heads/`, `tags/`, `remotes/` prefixes, and reflog entries follow documented rules; `<rev>:path` is refused). It also blocks agents from writing `refs/graft/` in every server mode and closes `SSHService/SSHUploadArchive` to agents; an `agent_rpcs` entry still naming `SSHService/SSHUploadArchive` now stops the server at startup and must be removed before upgrading. Separately, a revision passed as a branch name (e.g. `main~1`) now fails as not found rather than resolving via git revision rules, `UserCreateBranch` refused by access checks now returns `PermissionDenied` instead of `Internal`, and a contended repository lock in `CreateRepository`, `CreateFork`, `CreateRepositoryFrom*`, `CreateObjectPool`, `ReplicateRepository`, and backup restore now returns `FailedPrecondition` 'repository is already locked' instead of `Internal`.

— Every affected RPC named; one explicit pre-upgrade action specified. v0.4.0 v0.1.0

---

04 **`graft repo create` command for repository provisioning** NEW

84

Adds `graft repo create` CLI command and SDK `Repo.Create` method, letting operators create repositories with `--initial-commit` and `--branch` flags; the command is idempotent and supports both authoritative and shadow modes.

— Exact command, flags, and SDK method are directly usable. v0.4.0

05 **Storage checkpoints and garbage collection** NEW 78

Adds `[graft.storage]` checkpoints — a self-contained pack of a repository at a sequence — with hydration starting from the newest checkpoint and checkpoint builds never blocking a write; hydration always finishes, with rounds keeping progress and fetching 8 packs at a time, and a large checkpoint seeded in the background. Adds garbage collection of covered changeset packs, superseded checkpoints, packs of writes that died, and removed repositories, including leaks left by older versions. In authoritative mode, checkpoints now default to on (hourly) and garbage collection defaults to off; a `sync_timeout` of 55 minutes or more now requires `gc_grace` above `sync_timeout + gc_margin`, or the configuration fails at startup. `gitaly_housekeeping_tasks_total` gains a `skipped` status for `packed_refs`, so reftable events previously counted as failures now count as `skipped`.

— Config keys and defaults named; internal hydration mechanism lightly covered. v0.4.0 v0.3.0

06 **v0.3.0 upgrade migration for database and pack keys** BREAKING 75

Upgrading to v0.3.0 requires stopping every node, upgrading every node, then starting them: the first node to start migrates the database and pack keys change format, so a v0.2.x node cannot run on the migrated database; there is no downgrade past the migration without restoring the database and bucket from backup.

— Exact upgrade sequence and irreversibility are spelled out. v0.3.0

THINNER COVERAGE BELOW

07 **Pluggable repository storage, shadow/authoritative modes, changeset log**

NEW 56

Supports repository state in SQLite or PostgreSQL as a write-ahead log of changesets with compare-and-swap per reference, and pack storage on file, S3, GCS, or Azure via gocloud. Adds a changeset log that records every accepted reference update with its operation and actor, listable and watchable. Introduces shadow mode, which runs the new storage alongside classic Gitaly storage, and authoritative mode, where any node can serve and hydrate any repository from the database and object storage.

— Names backends and modes but gives no config or command detail. v0.1.0

08 **Review service with semantic approval and post-merge checks**

NEW 38

Introduces a review service with semantic approval rules (e.g. comment-only changes can require fewer approvals), post-merge verification with automatic bisect and revert, superfan monitoring agents, and tiered object caching.

— Names components only; no mechanism or configuration given. v0.1.0

09 **Windows support for CLI and SDK** NEW 24

Adds Windows support for the CLI and SDK.

— Bare statement with no scope or mechanism given. v0.3.1

└──▷ BREAKING ON UPGRADE

! A contended repository lock in CreateRepository, CreateFork, CreateRepositoryFrom*, CreateObjectPool, ReplicateRepository, and backup restore now returns FailedPrecondition 'repository is already locked' instead of Internal.

! UserCreateBranch refused by access checks now returns PermissionDenied instead of Internal.

! A revision passed as a branch name (e.g. `main~1`) now fails as not found when the branch is read, rather than resolving via git revision rules.

! `gitaly_housekeeping_tasks_total` gains a new `skipped` status for `packed_refs`; on reftable, events that previously counted as failures will now count as `skipped`.

! An `agent_rpcs` entry naming `gitaly.SSHService/SSHUploadArchive` now stops the server at startup — that entry must be removed before upgrading.

! Stop every node, upgrade every node, then start them: the first node migrates the database and pack keys change format; a v0.2.x node cannot run on the migrated database. There is no downgrade past the migration — going back requires restoring the database and bucket from a backup.

! `sync_timeout` of 55 minutes or more now requires `gc_grace` above `sync_timeout + gc_margin`; configurations that do not satisfy this fail at startup.

! New defaults in authoritative mode: checkpoints on (hourly), garbage collection off.

! `graft.land.batch.max_size` is now capped at 100; a larger value is rejected at startup.

! Conflict resolvers must now set `origin: "rule"` or `origin: "model"` on every resolved file; unmarked answers are treated as model resolutions, making former rule answers into proposals that land only if `verify.command` passes.

! A `command` strategy that calls a model must set `origin = "model"`.

! graft-resolve's rerere entries written by v0.1/v0.2 are never replayed; clear the store to reclaim them.

! `graft-resolve` and the server must be upgraded together: a new `graft-resolve` gives no model resolution to a server that does not announce the `proposal` (and `hold`) capability; an older `graft-resolve` against this server treats all answers as proposals subject to `verify.command`.

! SDK: `Land` now retries `Unavailable` up to 5 times with jittered backoff for merge and fast-forward lands; disable with `LandOptions.RetryUnavailable`.

WAS THIS USEFUL? 👍 👎

Google Gen AI JavaScript SDK

Feedback ⓘ v2.31.0

NOTES ↗ RANK

The Google Gen AI JavaScript SDK provides TypeScript and JavaScript interfaces to Gemini APIs and live model sessions.

> v2.31.0 extends the Live API with in-session context updates and new session config fields, adds `store` and `generation_output_config` to `GenerateContentConfig`, broadens `FunctionResultSubContent` to support audio, document and video types, and enables a single-value enum fix across all SDKs.

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

01 New GenerateContentConfig fields for output control NEW

80

Adds a `store` field to `GenerateContentConfig` (Gemini Developer API only) and a `generation_output_config` field, letting callers configure output behavior alongside other generation settings, e.g. `config: { generation_output_config: { ... }, store: true }`.

Configure output format alongside other generation settings using the new `generation_output_config` field in `GenerateContentConfig`.

typescript

```typescript
const response = await ai.models.generateContent({
  model: 'gemini-flash-latest',
  contents: 'Summarize this document.',
  config: {
    generation_output_config: { /* output config fields */ },
    store: true
  }
});
```

– Named config fields plus a runnable code example v2.31.0

02 Live API session config: context updates, labels, voice base

NEW

65

Adds `send_context_update` for in-session context updates during Live API sessions, available across GenAI SDKs. Adds a `labels` field to `LiveConnectConfig`, and a `base_voice` field to `PromptedVoice` for voice configuration.

– Names exact fields/method but no usage example or deeper mechanism shown v2.31.0

THINNER COVERAGE BELOW

03 **FunctionResultSubContent gains audio, document, video types**

NEW

45

`FunctionResultSubContent` now includes `AudioContent`, `DocumentContent`, and `VideoContent` as supported types.

— Names the new types but gives no usage detail v2.31.0

04 **Single-value enum fix flag for all SDKs** IMPROVED

40

Enables `fixes.openSingleValueEnumsOct2026` across all SDKs.

— Exact flag named but behavior and impact unexplained v2.31.0

WAS THIS USEFUL? 👍 👎

Google Gen AI Python SDK

Feedback ⓘ v2.33.0 NOTES ↗

RANK

The Google Gen AI Python SDK provides Python interfaces to Gemini model generation, streaming, tools, and embeddings.

> The Python Gen AI SDK's v2.33.0 release adds server-side content storage control, live session labelling, pluggable HTTP transport via httpx2, broader function-result content types, and Python 3.15 compatibility.

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

01 Server-side content storage flag in GenerateContentConfig NEW

75

Adds a `store` field to `GenerateContentConfig` (Gemini Developer API only) to control whether generate_content outputs are persisted server-side, set via types.GenerateContentConfig(store=True).

Persist generate_content outputs server-side (Gemini Developer API) by enabling the `store` flag in config.

python

```python
from google import genai
from google.genai import types

client = genai.Client(api_key='GEMINI_API_KEY')

response = client.models.generate_content(
    model='gemini-flash-latest',
    contents='Summarize the history of the internet.',
    config=types.GenerateContentConfig(store=True)
)
```

— Names exact config field and API scope with runnable example v2.33.0

02 Labels field for live connection sessions NEW

75

Adds a `labels` field to `LiveConnectConfig` for tagging live connection sessions, e.g. types.LiveConnectConfig(labels={'env': 'prod', 'team': 'search'}), useful for metadata like cost attribution or routing.

Tag a live session with metadata labels for cost attribution or routing.

```python
from google import genai
from google.genai import types

client = genai.Client(api_key='GEMINI_API_KEY')

async with client.aio.live.connect(
    model='gemini-flash-latest',
    config=types.LiveConnectConfig(labels={'env': 'prod',
'team': 'search'})
) as session:
    pass
```

— Names exact config field with runnable example v2.33.0

03 **httpx2 support via GOOGLE_GENAI_HTTP_CLIENT** NEW 75

The SDK now supports httpx2 as its HTTP client, selectable via the `GOOGLE_GENAI_HTTP_CLIENT` environment variable (e.g. `export GOOGLE_GENAI_HTTP_CLIENT=httpx2`), switching transport without changing application code.

Switch the SDK to httpx2 for HTTP transport without changing application code.

```
$ export GOOGLE_GENAI_HTTP_CLIENT=httpx2
```

— Names exact env var with runnable command v2.33.0

THINNER COVERAGE BELOW

04 **New content types for function result sub-content** NEW 35

Adds `AudioContent`, `DocumentContent`, and `VideoContent` as supported types in `FunctionResultSubContent`.

— Names types but gives no usage example or mechanism v2.33.0

05 **Python 3.15 compatibility** IMPROVED 15

The SDK adds compatibility with Python 3.15.

— Bare compatibility note with no further detail v2.33.0

WAS THIS USEFUL? 👍 👎

OpenAI TypeScript SDK

Feedback ⓘ v7.43.0 NOTES ↗ RANK

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

> v7.43.0 reworks the Live API Decisions input model: `DecisionInputMessage` becomes a union type, and Decisions questions and request inputs gain multi-part instructions, new message roles, and tool-history items — several of which are breaking for existing code.

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

01 **DecisionInputMessage becomes a union type** BREAKING 70

`DecisionInputMessage` is now a union type. Code that extends or implements the old interface must switch to the narrower `DecisionInputMessage.DecisionInputUserMessage` sub-type to keep compiling.

Update code that implements the old `DecisionInputMessage` interface to use the narrower `DecisionInputUserMessage` sub-type after the union change.

typescript

```
import type { DecisionInputMessage } from 'openai';

// Before (breaks in v7.43.0):
// const msg: DecisionInputMessage = { role: 'user', content:
'Hello' };

// After:
const msg: DecisionInputMessage.DecisionInputUserMessage = {
  role: 'user',
  content: 'Hello',
};
```

— Names exact type and migration path, includes before/after code v7.43.0

02 **Multi-part instructions and new input types for Decisions** 70

BREAKING

A Live response Decisions question's `instructions` field now accepts either a string or an array of text, image, or JSON parts, enabling richer multi-modal instructions. Decisions request inputs also now accept new message roles and tool-history items, broadening the input types supported in Live sessions. This is breaking for code that assumes `instructions` is always a string (e.g. question.instructions.trim()) or that

exhaustively switches over input types — callers must add a `typeof question.instructions === 'string'` guard and handle the array case.

Guard against the `instructions` field being an array before calling string methods, now that multi-part instructions are supported.

```typescript
if (typeof question.instructions === 'string') {
  console.log(question.instructions.trim());
} else {
  // instructions is an array of text, image, or JSON parts
  for (const part of question.instructions) {
    console.log(part);
  }
}
```

— Names the field and new types, gives a concrete guard pattern `v7.43.0`

└──▷ BREAKING ON UPGRADE

! `DecisionInputMessage` is now a union type — code that extends or implements the old interface must be updated to use `DecisionInputMessage.DecisionInputUserMessage` instead.

! A Live response Decisions question's `instructions` field can now be a string or an array; code such as question.instructions.trim() will throw when it receives an array — add a `typeof question.instructions === 'string'` guard and handle the array case.

! Decisions request inputs accept new message roles and tool-history item types — code that assumes it knows every input type (e.g. exhaustive switches or narrow type guards) may break.

WAS THIS USEFUL? 👍 👎

OpenAI Python SDK

Feedback ⓘ v3.37.0 NOTES ↗ RANK

The OpenAI Python SDK provides typed Python clients for OpenAI model APIs, streaming, and asynchronous requests.

> OpenAI Python SDK v3.37.0 expands the Decisions API's input flexibility, allowing richer structured content in both request inputs and question instructions — with breaking changes for strict type handling.

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

01 Expanded structured input types in Decisions API

BREAKING

82

Decisions request input types now include additional message roles and tool-history items for `DecisionInputMessageParam`, enabling richer agentic workflows, and in Live response models the question `instructions` field now accepts either a plain string or a list of structured text, image, or JSON parts. These are breaking changes: code doing string operations directly on `question.instructions` must add isinstance(question.instructions, str) guards to handle the new structured form, and exhaustive `match` / `if-elif` type checks over `DecisionInputMessageParam` must account for the new variants.

Guard against the new structured `instructions` form when processing Live response Decisions to avoid string-operation errors.

python

```python
if isinstance(question.instructions, str):
    text = question.instructions
else:
    # instructions is now a list of structured parts (text,
image, or JSON)
    text = ' '.join(
        part['text'] for part in question.instructions
        if part.get('type') == 'text'
    )
```

— Names exact types/field and gives a migration code guard. v3.37.0

└──▷ **BREAKING ON UPGRADE**

! In Live response models, `question.instructions` can now be a string or a list of structured parts — code that applies string operations directly to `question.instructions` will break; add isinstance(question.instructions, str) guards and handle the structured form.

! Decisions request input types now include additional message roles and tool-history items — exhaustive type checks (e.g. `match` / `if-elif` chains) over `DecisionInputMessageParam` will break if they do not account for the new variants.

WAS THIS USEFUL? 👍 👎

AWS Amazon Bedrock AgentCore Feedback ⓘ

Evaluations: Custom multi-dimension evaluators (2026-10-22) BLOG ↗ RANK

Amazon Bedrock AgentCore provides managed agent runtimes, memory, identity, gateways, observability, and evaluation services.

> Amazon Bedrock AgentCore Evaluations now supports custom multi-dimension LLM-as-a-judge evaluators that score a single target across several named dimensions in one judge call.

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

01 **Custom multi-dimension evaluators in AgentCore Evaluations** NEW

64

Amazon Bedrock AgentCore Evaluations adds custom multi-dimension evaluators: LLM-as-a-judge evaluators that rate one target on several named dimensions in a single judge call, each dimension carrying its own instruction and numerical or categorical scale, returning per-dimension results. These evaluators work across on-demand, online, and batch evaluation modes.

— Explains mechanism and modes but no exact API or config key given.

Evaluations: Custom multi-dimension eva…

WAS THIS USEFUL? 👍 👎

MotleyAI SLayer

Feedback ⓘ v1.4.0 NOTES ↗ RANK

An embeddable, expressive semantic layer for AI agents and humans

> SLayer's biggest window changes are per-caller model access control (TagFilteredStorage renamed to AccessFilteredStorage), Trino promoted to a Tier 1 database, and new anonymous opt-out usage reporting, alongside a run of breaking changes to storage fingerprinting, ambiguous-join handling, the MCP query schema, and query-backed model storage.

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

01 Per-caller model access control via AccessFilteredStorage

90

BREAKING

Adds TagFilteredStorage(backend, tags={...}, bypass=False) to wrap a storage backend and give each caller a filtered view: callers see only untagged models or models sharing their `access_tags`, with hidden models propagating to dependent models and memories. Adds an `access_tags` field on models; only bypass callers may set or change `access_tags` (enforced by `AccessTagsEditError`), and writes conflicting with hidden content raise `HiddenContentConflictError`. A `narrowed_access` warning is emitted into `create_model` / `edit_model` MCP replies when a saved model reads a more narrowly tagged model. In the next release, `TagFilteredStorage` was renamed to `AccessFilteredStorage` and moved from `slayer.storage.tag_filtered` to `slayer.storage.access_filtered` with no compatibility alias, so any existing import or reference breaks on upgrade.

– Full mechanism, class signature, error types, and breaking rename v1.3.0 v1.2.0

02 Trino promoted to Tier 1 database NEW

88

Adds Trino as a Tier 1 database, installable via `pip install 'motley-slayer[trino]'`, with support for date shifts, relative date ranges, week buckets, time spines, query timeouts, and HTTPS logins on any port; the same improvements also apply to Presto and Athena.

– Install command plus concrete feature list for a named backend v1.2.0

03 Anonymous opt-out usage reporting NEW

86

Adds anonymous, aggregated daily usage reporting to processes started by the `slayer` command (`slayer serve`, `slayer mcp`, `slayer query`, Docker image); it never reports queries, data, model or column names, credentials, or error messages — only counts and fixed tokens. Provides `slayer usage-reporting disable` and `slayer usage-reporting show` subcommands, plus `DO_NOT_TRACK=1` and `SLAYER_USAGE_REPORTING=off` environment variables to control and inspect it.

— Names exact subcommands and env vars, fully actionable `v1.4.0`

04 **Expanded formula and query expression syntax** IMPROVED

75

Supports count(distinct x) as an alias for count_distinct(x), method-call aggregate syntax such as order_items.count(*), and date-part aliases `isodow`, `day_of_year`, and extract(month from order_date) in formula expressions. Measures can be named after a transform (e.g. `change`) without a distinct name, an order column can be specified as `{"name": "<expression>"}`, and `partition_by=[month(order_date)]` is supported over a monthly time dimension. A refinement can restate a saved measure by name, and division by zero in a formula now returns NULL on every backend instead of raising an error.

— Enumerates every added syntax form verbatim `v1.3.0`

05 **Removal of stored backing_query_sql** BREAKING

75

Query-backed models no longer store `backing_query_sql`; run a `dry_run` query to see the SQL. The v15 migration drops `backing_query_sql` from stored models.

— Gives exact migration version and the replacement workflow `v1.2.0`

06 **Join removal in edit_model** NEW

70

Adds `remove.join_edges: [{target_model, name, join_pairs}]` parameter to `edit_model` to remove one of several joins to the same target table.

— Exact parameter shape given but thin on surrounding context `v1.2.0`

07 **Full ingestion of foreign-key cycle schemas** NEW

70

Schemas with foreign-key cycles now ingest fully, with every FK becoming a join; when two FKs point at the same table, joins are named after their columns (e.g. `billing_address.city` and `shipping_address.city` from `orders`).

— Clear mechanism and example, but automatic rather than invoked `v1.2.0`

08 **Cross-model aggregates without a join path** NEW

65

Adds an `unrelated_dataset` warning flag on result rows when an aggregate from a model with no join path to the query root is computed inline (e.g. count(events.id) / count(*) rooted at `customers`). Extends `recommend_root_model` to accept unrelated-model items and marks them via a new `unrelated: bool` field on `ItemPath`.

— Named fields and flags but no direct command to invoke `v1.3.0`

09 **Storage backend fingerprinting and caching changes** BREAKING

65

Adds StorageBackend.cache_identity() method on storage backends; YAML and SQLite backends now compute exact fingerprints. The default graph_fingerprint() is now None (never reused), changing caching behaviour for storage backends that previously relied on fingerprint reuse.

— Names the method and behaviour change but no migration steps given v1.2.0

10 **MCP query argument schema change** BREAKING 60

The `query` argument's JSON schema in MCP changes from `anyOf` to `oneOf` (string = saved model name, object = query, array = stages).

— Precise schema change but no example of a failing call v1.2.0

THINNER COVERAGE BELOW

11 **Ambiguous join detection on save** BREAKING 55

Raises `DerivedColumnAmbiguousError` at save time when a derived column or model filter path crosses an ambiguous join; this is now an error rather than being silently allowed.

— Names the error type but no detail on resolving it v1.2.0

╰──▷ BREAKING ON UPGRADE

! `TagFilteredStorage` is renamed to `AccessFilteredStorage` and moved from `slayer.storage.tag_filtered` to `slayer.storage.access_filtered` with no compatibility alias — any import or reference to `TagFilteredStorage` or `slayer.storage.tag_filtered` will break on upgrade.

! Query-backed models no longer store `backing_query_sql`; run a `dry_run` query to see the SQL. The v15 migration drops `backing_query_sql` from stored models.

! Saving a derived column or model filter whose path crosses an ambiguous join is now an error (`DerivedColumnAmbiguousError`).

! The `query` argument's JSON schema in MCP changes from `anyOf` to `oneOf` (string = saved model name, object = query, array = stages).

! The default graph_fingerprint() is now None (never reused), changing caching behaviour for storage backends that previously relied on fingerprint reuse.

WAS THIS USEFUL? 👍 👎

siliconscope

Feedback ⓘ v4.7.0 NOTES ↗ RANK

Sudoless Apple Silicon system monitor (native SwiftUI GUI) with ANE / Media Engine / memory-bandwidth tracking

> SiliconScope's four releases this window add deep per-card NVIDIA GPU monitoring with per-process account tracking on Linux, Docker container visibility (including a rootless agent install path), and a major network-efficiency overhaul that cuts agent traffic via persistent TLS connections and gzip compression, alongside new NVMe SMART reporting, per-interface network breakdown, Tailscale/VPN and Intel Mac support, and a breaking change excluding VPN tunnel interfaces from traffic totals.

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

01 Docker container visibility and rootless agent install NEW

95

Fleet now shows Docker containers per machine, including restart counts (containers restarted 400+ times shown in amber), ComfyUI queue depth, and which process holds VRAM named by script and container instead of generic `python3`. Cloud servers without a GPU also display their Docker containers, and each Fleet tile now shows a container count. The agent installer can run as a dedicated non-root user via the `SSCOPE_USER` and `SSCOPE_DOCKER_HOST` environment variables, pointing at a rootless Docker socket proxy so the agent never needs the root-owned Docker socket, e.g. `SSCOPE_USER=sscope SSCOPE_DOCKER_HOST=unix:///run/user/1000/docker.sock bash install.sh`.

Run the Fleet agent installer as a dedicated non-root user with a rootless Docker socket proxy, so the agent never needs access to the root-owned Docker socket.

```
$ SSCOPE_USER=sscope
SSCOPE_DOCKER_HOST=unix:///run/user/1000/docker.sock bash
install.sh
```

— Runnable install command with named env vars v4.5.0

02 Per-card NVIDIA GPU monitoring and account tracking in Fleet

NEW

83

Fleet now reports NVIDIA Linux GPU boxes in depth: memory-controller utilisation, SM and memory clocks versus their maxima, P-state, fan speed, NVENC/NVDEC activity, and clock-throttle reasons (power cap, thermal slowdown, power brake). On multi-GPU machines each card is shown individually — per-card load, memory, temperature, charts,

process lists and independent history, with the overview tile charting up to two cards and the sidebar showing every card's load. The Linux agent (v1.5.0) matches each GPU process to its card by UUID, lists processes that span multiple cards under each card with per-card memory, and shows the account the process runs as.

On a shared Linux box with multiple NVIDIA GPUs, identify which user account is holding GPU memory under a specific card.

> 📍 In Fleet, select the target machine, then click the chip for the GPU card of interest — the Processes view lists each job with its owning account and the memory it holds on that card.

— Detailed mechanism and metrics but no CLI command `v4.7.0` `v4.5.0`

---

03 **Network traffic reduction via persistent connections and gzip**

IMPROVED

81

Switches to one persistent TLS connection per machine instead of a new connection per poll, cutting traffic from ~106 KB/min to ~29 KB/min while the window is open. Poll replies are now gzip-compressed — 7.7 KB to 2.4 KB from a Mac agent, 1.5 KB to 0.6 KB from a Linux agent — requiring agent update to 1.4.0. All polling halts and connections close when the Fleet window is closed or minimised, dropping background network usage to zero, and polling resumes immediately when the window is reopened, showing a gap in charts rather than interpolating across the break.

— Concrete numbers and mechanism, no direct user command `v4.5.1`

---

04 **NVMe SMART readings and lifetime totals on SSD card** NEW

73

Adds NVMe SMART readings for the internal SSD — remaining rated endurance, temperature, spare capacity, media errors, and each critical warning by name — shown on the dashboard Disk card and the SSD menu. Also adds lifetime and since-boot read/write totals sourced from the storage driver's own counters, covering activity before SiliconScope launched.

Review NVMe SMART health data — remaining endurance, temperature, spare capacity, media errors, and named critical warnings — without needing a separate SMART utility.

> 📍 In SiliconScope, open the Disk card on the dashboard or click the SSD menu; SMART readings (endurance, temperature, spare capacity, media errors, critical warnings) are shown read-only alongside since-boot and lifetime read/write totals.

— Detailed metrics, UI path but no command `v4.6.0`

---

05 **VPN tunnel interfaces excluded from network totals**

BREAKING

65

`utun` interfaces used by Tailscale and WireGuard are no longer counted toward aggregate network totals; after upgrading to Mac agent 1.4.1, dashboards or alerts based on aggregate throughput will show lower numbers.

— Clear before/after but no migration command `v4.6.0`

06 **Per-interface network breakdown and address display** NEW

65

The Network section lists each interface (Ethernet, Wi-Fi) with its own download and upload totals, leading with the default-route interface. Expanding an interface row shows full copyable IPv4 and IPv6 addresses, and checking More addresses reveals VPN and tunnel interfaces.

Check which physical interface is carrying the default route and compare its true throughput against Wi-Fi — useful when a VPN was previously inflating totals.

📍 In SiliconScope, open the Network section; the interface carrying the default route is listed first, each showing its own download and upload. VPN `utun` interfaces are excluded from all totals.

Confirm all local network addresses (including VPN tunnels) for a monitored host without triggering any online lookup.

📍 In the Network section for the target host, expand the interface row to see the full copyable IPv4 and IPv6 addresses; check More addresses for VPN and tunnel interfaces.

— Clear navigation steps, no API/command `v4.6.0` `v4.7.0`

THINNER COVERAGE BELOW

07 **Agent install support for older systemd hosts** NEW 55

Synology NAS and other hosts running older systemd (e.g. DSM 7 with systemd 219) are now supported: the installer creates the agent state directory itself.

— Names specific OS/systemd version but no command shown `v4.5.0`

08 **Intel Mac fan and temperature sensors in Fleet** NEW

45

Intel Mac fans and temperatures are now reported in Fleet, with CPU die sensors named by their documented meaning, and the fanless label now appears only for genuinely fanless hardware.

— Describes change but no steps or command `v4.5.0`

09 **Tailscale and VPN machine connectivity in Fleet** NEW

45

Machines added by a Tailscale (100.x) or VPN address now connect to Fleet, with certificate pinning on first use preserved.

— States capability but no setup steps v4.5.0

10 **Memory bandwidth label corrected to 'Mem BW'** IMPROVED

35

Renames the ANE 0% · BW 271 GB/s label to Mem BW to correctly identify the metric as whole-chip memory bandwidth, not Neural Engine traffic.

— Simple label fix, no further detail v4.5.1

11 **Install command shown on empty Fleet page** IMPROVED 35

The empty Fleet page now shows the install command with a copy button and links to the Fleet manual.

— UI path given but no command text itself v4.5.0

12 **SiliconScopeCore library usage guide** NEW 30

Publishes a Using SiliconScopeCore as a library guide for embedding the measurement layer into third-party tools.

— Documentation pointer only, no details given v4.5.0

└──▷ **BREAKING ON UPGRADE**

! `utun` interfaces (Tailscale, WireGuard) are no longer counted in network totals; dashboards or alerts based on aggregate throughput will show lower numbers after upgrading to Mac agent 1.4.1.

WAS THIS USEFUL? 👍 👎

skillhub

Feedback ⓘ v0.2.23 NOTES ↗ LATEST STABLE

Self-hosted, open-source agent skill registry for enterprises. Publish & version skill packages, govern with RBAC and audit logs, deploy on-premise with Docker or Kubernetes.

SkillHub v0.2.23 adds Skill avatars, single-use team invitations, standalone visibility review, draft authoring APIs, label management via `skill:publish` tokens, and local sign-in/self-registration settings.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.2.23 https://github.com/iflytek/skillhub.git
# already have the repo? check out this version:
$ git checkout v0.2.23
```

Adds draft CRUD, file management, and structure-validation APIs as the first layer of the authoring platform.

Allows API tokens with `skill:publish` scope to attach and detach Skill labels, retaining resource permission checks.

Adds independent, reviewed visibility changes for published Skills — audience expansion requires review; restriction takes effect immediately.

Adds local sign-in and self-registration settings, plus initial role grants for newly created external accounts with verified emails.

NEW DETECTION Adds Skill avatars with staged uploads and configurable avatar moderation; avatar and package changes are reviewed together when submitted in the same update.

+2 more

NEW INTEGRATION Native DeepSeek Harness plugin source

NEW INTEGRATION Feishu sign-in provider

NEW INTEGRATION DingTalk sign-in provider

└──▷ BREAKING ON UPGRADE

! The Server applies Flyway migrations V68–V75 (authentication settings, draft authoring, avatars, invitations, visibility review); schema rollback requires restoring the matching pre-upgrade PostgreSQL database and local Skill storage backups taken before upgrading.

▾ 1 more release in this issue · 2026-10-10 → 2026-10-19

v0.2.22 NOTES ↗ STABLE

Adds Organization and `SUPER_ADMIN` APIs, global Skill lifecycle controls, versioned IdP adapters, built-in Skills, and Mermaid diagram rendering.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.2.22
https://github.com/iflytek/skillhub.git
# already have the repo? check out this version:
$ git checkout v0.2.22
```

Adds membership-scoped Organization and Login Connection read APIs, plus a `SUPER_ADMIN` organization-creation API.

Adds global Skill promotion updates, revocation review, direct revocation, retained audit history, and safe re-promotion.

Adds versioned identity-provider adapter contracts.

Publishes `orca-replay`, `ledger-tasks-yylo`, and `triage-nda` as built-in Skills through pinned immutable artifacts.

Renders fenced Mermaid blocks as diagrams, with source fallback when rendering fails.

+1 more

└──▷ BREAKING ON UPGRADE

- ! API tokens belonging to a merged secondary account are revoked after the merge; affected automation must use newly issued tokens.
- ! Flyway migrations V66-V67 run on upgrade — back up PostgreSQL before upgrading.
- ! When upgrading to Scanner 2.1, the coordinated Server and Scanner rollout must be completed before enabling AI Defense.

WAS THIS USEFUL? 👍 👎

llm-sandbox

Feedback ⓘ 0.3.46 NOTES ↗ RANK

Lightweight and portable LLM sandbox runtime (code interpreter) Python library.

> llm-sandbox 0.3.46 adds four new daemonless sandbox backends for running LLM-generated code without a Docker daemon, covering Linux, Windows, and WSL2 execution paths, plus a backend registration mechanism for discovering adapter capabilities at runtime.

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

01 Bubblewrap backend for daemonless Linux sandboxing NEW

93

Adds `backend="bubblewrap"` to `SandboxSession` for daemonless local Linux execution using user namespaces and a configurable seccomp syscall denylist, with networking disabled by default and a private writable workspace. Requires Bubblewrap 0.8.0+ and system libseccomp.

Run LLM-generated Python in a namespace-isolated sandbox on a Linux host that has no Docker daemon — networking is off and the workspace is private by default.

python

```python
from llm_sandbox import SandboxSession

with SandboxSession(backend="bubblewrap", lang="python") as
session:
    result = session.run("print('Hello from a local Linux
sandbox!')")
    print(result.stdout)
```

— Named flag, version requirements, and a runnable example. 0.3.46

02 WSL2 Bubblewrap backend for Windows-to-Linux execution NEW

83

Adds experimental `backend="wsl2-bubblewrap"` to `SandboxSession` for running Linux tools from a Windows Python application inside an explicitly selected WSL2 distribution via the `distribution` parameter.

Run Linux tools from a Windows Python application by routing execution through an explicitly named WSL2 distribution with Bubblewrap isolation.

```python
from llm_sandbox import SandboxSession

with SandboxSession(
    backend="wsl2-bubblewrap",
    distribution="Ubuntu",
    lang="python",
) as session:
    print(session.run("import platform;
print(platform.system())").stdout)
```

— Named parameter and a runnable cross-platform example. 0.3.46

---

03 **NsJail backend for daemonless Linux sandboxing** NEW

71

Adds `backend="nsjail"` to `SandboxSession` for daemonless local Linux execution using NsJail 3.6 with Kafel-based native seccomp policies and the new mount API.

— Named flag and mechanism, but no runnable example. 0.3.46

---

04 **Windows MXC backend for native Windows execution** NEW

68

Adds experimental `backend="windows-mxc"` to `SandboxSession` for native Windows Python and JavaScript execution through a PSEC-capable trusted helper; the helper binary is selected with `mxc_path`.

— Named flags given but experimental and no example shown. 0.3.46

THINNER COVERAGE BELOW

05 **Backend registration and capability discovery** NEW 30

Adds backend registration and capability discovery so additional adapters can be registered and their supported features inspected at runtime.

— Described only in prose with no named API or example. 0.3.46

WAS THIS USEFUL? 👍 👎

◆ AI / LLM Gateways

cc-router

Feedback ⓘ v6.5.0 NOTES ↗ RANK

A local LLM gateway with a GUI desktop app that bundles scattered Token Plans, Coding Plans, and API quotas into one virtual Plan for Claude Code and other AI tools.

> cc-router's biggest change this window is time-of-day model slot switching with a route-map preview for Virtual models, alongside a breaking restructuring of the OpenAI-style alias tiers and support for custom Jev/Clef decision-model endpoints.

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

01 **Custom System One decision-model endpoint** NEW 95

Adds a Custom (System One decision model) option in the Provider dropdown's Custom group, letting you supply a Base URL, endpoint path, API Key, and Jev model to connect any LAN or third-party Jev/Clef endpoint to `model-jev` with no protocol translation, authenticating via `Authorization: Bearer` or `x-api-key`. The terminal UI new-subscription wizard gains the same option with a single Jev model field. When a `model-jev` request finds a subscription with neither a Jev model nor an example model, it skips that subscription and tries others, returning 503 with a reason instead of a blanket 400 if all are unavailable.

Connect a self-hosted Ollama instance on your LAN to `model-jev` so all routing through that slot uses your local model.

📍 In the cc-router UI, go to Subscriptions › Add Subscription, open the Provider dropdown, select Custom (System One decision model) under the Custom group, fill in Base URL (e.g. `http://192.168.1.50:11434`), endpoint path, API Key (any value if upstream skips verification), and the required Jev model name, then save.

— Names every field, auth header and status code, with a runnable example v6.4.0

02 **Time-of-day model slot switching** NEW 85

Each subscription slot now supports up to 16 ordered time rules — by day (Every day / Weekdays / Weekend / Custom) and start/end time to the half-hour, including cross-midnight spans — each with its own thinking effort, configured in the Model slots card on the subscription edit page. The terminal UI Subscriptions page shows the active model for each slot at the current moment, with a Timed badge on slots that have time rules.

Route a slot to a cheaper night-time model automatically, leaving it open to other subscriptions during off-rule hours.

📍 In the subscription edit page, open the Model slots card for the target slot. Set the default model to None (unavailable outside rules). Add a time rule: Days = Weekdays, Start = 22:00, End = 06:00, Model = <night-model>. The live weekly preview on the right will confirm coverage; Copy to other slots to apply the same schedule to the remaining three slots.

— Names exact config location and rule granularity, with a worked example v6.5.0

---

03 **OpenAI alias tier restructuring** BREAKING 80

Adds `gpt-*-astra` alias tier mapping to `model-fable`, expanding OpenAI-style tiers from three to four (astra / sol / terra / luna). Remaps `gpt-*-sol` to `model-opus` (previously `model-fable`), `gpt-*-terra` to `model-sonnet` (previously `model-opus`), and `gpt-*-luna` to `model-haiku` (previously `model-sonnet`), each moving down one tier. Requests using the now-unrecognized `gpt-5.6`, `gpt-5.5`, `gpt-5.4`, or `gpt-*-mini` aliases are routed to `model-fallback` instead of being rejected. The `GET /v1/models` response now lists 32 entries: Claude aliases in two versions per tier (e.g. `claude-opus-5-5`, `claude-opus-6`) and OpenAI aliases in one per tier (e.g. `gpt-6-sol`).

— Full mapping and endpoint change named, but no migration command v6.3.1

---

04 **Virtual models route map with time preview** NEW 65

The Virtual models page becomes a route map, expanding each virtual model through its subscriptions to individual time rules, with the currently matching row highlighted. Dragging View time to any half-hour in the week previews routing, with unavailable spans shown in red.

Check how requests will be routed at an arbitrary future time without waiting for the clock to reach it.

📍 Open the Virtual models page (now a route map). Drag View time at the top to any half-hour in the week — for example, Saturday 03:00 — and observe which time rule row is highlighted per subscription; spans with no usable subscription are shown in red.

— Clear UI mechanism and navigation, no deeper technical detail v6.5.0

05 **Empty model slots no longer routed** BREAKING — 65

Subscriptions with an empty model slot are no longer called: requests that previously reached the upstream with an empty model name (relied on by local services such as llama.cpp) are now skipped, and a 503 is returned if no other subscription is usable. Filling in any model name for the affected slot restores the previous behavior.

– Explains before/after and the fix but no step-by-step example v6.5.0

06 **Auto-update check toggle** NEW — 60

Adds a Check for updates automatically toggle (on by default) at Settings → General → App updates, also accessible from the Updates page, letting updates be checked on demand instead of at launch.

Disable the startup network call for update checks in an air-gapped or controlled environment.

📍 Go to Settings → General → App updates → Check for updates automatically and turn the switch off. cc-router will no longer contact the update source at launch; updates can still be checked on demand from the Updates page.

– Exact settings path given but mechanism is simple v6.5.0

THINNER COVERAGE BELOW

07 **Sign-in token visible in Web UI settings** IMPROVED — 50

Adds a Sign-in token row under Settings → Web & TUI → Web UI, visible whenever Login required is on, so the token can be copied without navigating to the Security & Access tab.

Find the sign-in token for your Web UI without digging through the Security & Access tab.

📍 In the cc-router UI, go to Settings → Web & TUI → Web UI. Ensure Login required is on. Copy the Sign-in token value shown below the access URLs.

– Exact settings path and condition given, with steps v6.3.1

08 **Traditional Chinese UI language** NEW — 40

Adds Traditional Chinese (繁體中文) as a UI language, selectable at Settings → General → UI language, covering desktop UI, web UI, terminal UI, tray menu, and provider names.

– Names the setting path but no usage example given v6.5.0

└──▷ BREAKING ON UPGRADE

! Subscriptions with an empty model slot are no longer called: requests that previously reached the upstream with an empty model name (relied on by local services such as llama.cpp) are now skipped, and a 503 is returned if no other subscription is usable. Fill in any model name for the affected slot to restore the previous behavior.

! `gpt-*-sol` now maps to `model-opus` (was `model-fable`); `gpt-*-terra` now maps to `model-sonnet` (was `model-opus`); `gpt-*-luna` now maps to `model-haiku` (was `model-sonnet`) — clients using these aliases will silently hit a different model tier after upgrading.

! `gpt-5.6`, `gpt-5.5`, `gpt-5.4`, and `gpt-*-mini` are no longer recognized; requests using these names are routed to `model-fallback` — update them to the tier names `gpt-*-astra`, `gpt-*-sol`, `gpt-*-terra`, or `gpt-*-luna`, or use virtual names such as `model-opus` directly.

WAS THIS USEFUL? 👍 👎

dario

Feedback ⓘ v6.15.0 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's biggest addition this window is `--family-fallback`, which routes rate-limited model requests to a working Claude model instead of returning 429, refined across three releases for safety and completeness; it also gained MiniMax regional setup presets and an updated bundled Claude Code template.

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

01 Family fallback routing for rate-limited models NEW

95

The `--family-fallback` flag (env `DARIO_FAMILY_FALLBACK`, config key `familyFallback`) automatically routes requests for a parked model family to a specified Claude model when rate limits exhaust the family's credits, e.g. `--family-fallback=fable=claude-opus-5-5` sends Fable requests as Opus; the response includes an `x-dario-family-fallback` header naming the originally requested model, and `dario doctor` reports the `familyFallback` configuration when set. A safety guard leaves model names containing non-ASCII characters, spaces, control characters, line breaks, or names over 256 characters unchanged, echoing the original name in the same header and log. Parked seats now return a local `429` with the reset time from the named bucket (or route via `--family-fallback`) instead of making a redundant upstream round trip, and the fallback now also re-serves the request that originally triggered the `429` itself (previously dropped or forwarded to `--pool-fallback`), marking it `x-dario-family-fallback` like any other switched request.

Prevent Fable requests from returning 429 when credits are exhausted by falling back to Opus automatically.

```
$ dario --family-fallback=fable=claude-opus-5-5
```

Set the family fallback persistently via config so all dario invocations use it without a flag.

```yaml
familyFallback: fable=claude-opus-5-5
```

— Flag, env var, config key, header, and runnable example all given. v6.15.0 v6.14.1 v6.14.0

02 **Bundled Claude Code template updated to 2.1.304** IMPROVED

68

Adds `DeferredToolPlaceholder` and `ToolSearch` to the bundled tool list and `advanced-tool-use-2025-11-20` to the bundled `anthropic_beta` set. Updates the bundled template label and `user-agent` to Claude Code 2.1.304 (claude-cli/2.1.304 (external, sdk-cli)), marks it verified against 2.1.304, and suppresses the startup warning about 2.1.303. Preserves `defer_loading: true` on `DeferredToolPlaceholder` so deferred tools are not sent as normal tools.

— Names concrete tools, beta flag and user-agent string but no user command. v6.13.22

THINNER COVERAGE BELOW

03 **MiniMax backend regional setup presets** NEW

52

Adds regional setup presets for MiniMax API-key backends: the `minimax` preset selects MiniMax-M3.1-Flash on a direct MiniMax endpoint, or routes to a custom gateway using that gateway's own model ID.

— Names the preset and behavior but gives no runnable command. v6.13.29

WAS THIS USEFUL?  

Mozilla AI Otari

Feedback ⓘ v0.32.0 NOTES ↗ RANK

Open-source, OpenAI-compatible LLM gateway you run yourself. One endpoint for 40+ providers, with virtual keys, budgets, and usage tracking.

> Otari v0.32.0 adds DeepSeek search/fetch adapters, per-tool `allowed_tools` access control for search tools, flat-rate `price_per_request` pricing, and `safety_identifier` tracking across usage rows and inference calls, alongside breaking removals of `provider_user_identifier_enabled` and `b-hybrid` mode.

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

01 **safety_identifier tracking across usage and inference** NEW

70

Adds a `safety_identifier` column to usage rows, storing the end-user identifier on each usage row; uses the caller's `safety_identifier` as the end-user tag in inference, sending it on OpenAI chat completion requests, and carries the end-user identifier into image describe calls.

— Field and propagation paths named, no call example shown v0.32.0

02 **Access control for search tools via allowed_tools** BREAKING

65

Search tools now require their own `allowed_tools` access list, giving each search tool independent access control; this is a breaking change from prior behavior where no such list was required.

— Config field named, but no migration example given v0.32.0

THINNER COVERAGE BELOW

03 **Server-Timing split for request duration** NEW 55

Splits a response's duration into Otari's time and the provider's time, exposed in Server-Timing response headers.

— Header named, readers can inspect directly v0.32.0

04 **DeepSeek search and fetch adapters** NEW 50

Adds DeepSeek search and fetch adapters via `any-search` and `any-fetch`, extending provider support for search/fetch tools.

— Names adapters but no config example v0.32.0

05 **Guardrail and Claude Code session tracing** NEW 50

Records each guardrail check an agent's hook makes on its Otari session, and reads Claude Code's own spans into a session's turns and tools.

— Describes behavior, no config or command given v0.32.0

06 **Flat-rate pricing via price_per_request** NEW 45

Adds a `price_per_request` pricing field to price a call at a flat USD rate.

— Field named but no usage example v0.32.0

07 **Organization search and fetch keys on library providers** 40

BREAKING

Adds organization search and fetch keys on library providers; this is a breaking change to existing library provider key behavior.

— Thin description, no detail on what changed v0.32.0

08 **Model catalog metadata** IMPROVED 40

Lists each model's name, tool support, and modalities in the model catalog.

— Fields named but no endpoint or path given v0.32.0

09 **Removal of provider_user_identifier_enabled** BREAKING 40

The `provider_user_identifier_enabled` switch has been removed from inference configuration.

— Exact config key named, no replacement guidance given v0.32.0

10 **Member-assigned federation rules** NEW 30

Lets members assign federation rules to their own keys.

— Bare description, no mechanism named v0.32.0

11 **Removal of b-hybrid mode** BREAKING 20

The `b-hybrid` mode has been removed.

— Bare removal notice, no migration detail v0.32.0

└──▷ BREAKING ON UPGRADE

! The `provider_user_identifier_enabled` switch has been removed from inference configuration.

! The `b-hybrid` mode has been removed.

! Organization search and fetch keys on library providers changed behavior (marked BREAKING in release notes).

! Search tools now require an `allowed_tools` access list of their own (marked BREAKING in release notes).

WAS THIS USEFUL?  

opencode-models-discovery

Feedback ⓘ v2.0.0 NOTES ↗

RANK

A universal OpenCode plugin for dynamic model discovery with flexible configuration for OpenAI-compatible providers.

> opencode-models-discovery shipped a unified TUI discovery console, Integration-backed credential resolution for OpenCode v2 providers, and host-managed per-provider discovery caching, alongside a shared V1/V2 discovery core, expanded provider metadata, and a metadata-driven replacement for heuristic reasoning detection.

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

01 **Host-managed discovery caching for OpenCode v2** NEW

93

Adds `cache.enabled` and `cache.ttlSeconds` under `providers.<id>.settings.modelsDiscovery` in `opencode.json` to enable provider-scoped host-managed discovery caching for OpenCode v2, storing raw provider models and enrichment results through the host `ctx.storage` API; caching is disabled by default and uses a 24-hour TTL when `ttlSeconds` is omitted.

Enable per-provider discovery caching for an OpenCode v2 gateway provider so repeated startups avoid hitting the upstream models endpoint.

```json
{
  "plugins": ["opencode-models-discovery@2.0.0"],
  "providers": {
    "gateway-custom": {
      "name": "Gateway Custom",
      "package": "@opencode/ai/providers/openai-compatible",
      "settings": {
        "baseURL": "https://gateway.example.com/v1",
        "apiKey": "{env:GATEWAY_API_KEY}",
        "modelsDiscovery": {
          "enabled": true,
          "cache": {
            "enabled": true,
            "ttlSeconds": 86400
          }
        }
      }
    }
  }
}
```

— Exact config keys, defaults, and runnable config example. v1.9.0

02 **Integration-backed credential resolution for providers** NEW

91

Adds `settings.integrationID` config key under a provider's `settings` to reuse a credential from another OpenCode v2 Integration instead of duplicating an API key, plus Integration-aware credential resolution letting providers omit `settings.apiKey` from `opencode.json` and authenticate via the `/connect` workflow, with resolved keys never written to `opencode.json`, the discovery cache, diagnostics, or logs. The plugin also reacts to OpenCode v2 credential updates, credential switching, Integration updates, and configuration updates, refreshing discovery and runtime auth without an OpenCode restart.

Authenticate a custom provider without embedding an API key in config — run `/connect` in the TUI after adding the provider block.

```json
{
  "plugins": ["opencode-models-discovery@2.0.0"],
  "providers": {
    "deepseek-custom": {
      "name": "DeepSeek Custom",
      "package": "@opencode-ai/ai/providers/openai-compatible",
      "settings": {
        "baseURL": "https://api.deepseek.com",
        "modelsDiscovery": {
          "enabled": true,
          "endpoint": "/models"
        }
      }
    }
  }
}
```

Reuse an already-connected Integration credential across a second provider to avoid entering the same API key twice.

```json
{
  "providers": {
    "deepseek-alt": {
      "name": "DeepSeek Alt Endpoint",
      "package": "@opencode-ai/ai/providers/openai-compatible",
      "settings": {
        "baseURL": "https://alt.deepseek.com",
        "integrationID": "deepseek-custom",
        "modelsDiscovery": {
          "enabled": true,
          "endpoint": "/models"
        }
      }
    }
  }
}
```

— Named config key and workflow with working config examples. v1.10.0

03 **Unified TUI discovery console** NEW 90

Adds `/models-discovery-console` command combining provider→model→details browsing, cache status inspection, force refresh, per-provider cache clearing, and full

cache clearing, plus `/models-discovery-console status --provider <provider-id>` to filter the console browser to a single provider.

Inspect discovery status and cache state for a single provider in the TUI console.

```
$ /models-discovery-console status --provider deepseek-custom
```

— Exact command and flag given, directly runnable. v1.10.0

04 **Structured model discovery status tool** NEW INTEGRATION IMPROVED

80

The `models_discovery_status` Agent Tool (and console) exposes structured provider identity, model counts, metadata, limits, capabilities, variants, and cache state; model details are now shown in separately inspectable sections — limits, capabilities, modalities, reasoning support, attachments, and variants — instead of a single unformatted report, and the tool was expanded to report context/output limits and cache status. A `rawCache: true` parameter includes cached raw models and enrichment results in the response.

Inspect cached raw discovery data and enrichment results for a provider via the agent tool, without triggering a fresh discovery run.

```json
{
  "details": true,
  "rawCache": true
}
```

— Names tool and parameter but not exact invocation syntax. v1.10.0 v1.9.0

THINNER COVERAGE BELOW

05 **Shared V1/V2 discovery and projection core** IMPROVED

50

Introduces a shared discovery and projection core used by both OpenCode host generations, providing consistent raw model validation, normalization, filtering, classification, smart naming, duplicate-name disambiguation, model-info enricher resolution, limit normalization, and safe context-only output defaults.

— Lists internal mechanisms but no direct user action. v1.9.0

06 **Provider metadata support for Bifrost and OmniRoute**

NEW INTEGRATION NEW

48

Adds support for Bifrost `reasoningEffort` variants and context-only model limits, and for OmniRoute context-only limits and service-tier variants, in provider metadata.

— Names both providers and fields but no config example. v1.9.0

07 **Workspace-aware discovery operations** IMPROVED 30

Discovery status, refresh, and cache operations now use the active OpenCode workspace location, fixing behavior in multi-workspace sessions.

— States the fix but gives no reproduction steps. v1.10.0

08 **Metadata-driven reasoning capability detection** BREAKING 30

Removes heuristic name-based reasoning capability guessing, replacing it with explicit metadata-driven detection.

— States the change but no migration guidance given. v1.9.0

NEW INTEGRATION models.dev catalog integration

v2.0.0 NOTES ↗ LATEST STABLE

v2.0.0 promotes OpenCode v2 as the stable integration, adding host-managed credentials, model overrides, TUI commands, and agent tools.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.0.0 https://github.com/yuhp/opencode-models-discovery.git
# already have the repo? check out this version:
$ git checkout v2.0.0
```

Adds `providers.<provider-id>.settings.modelsDiscovery` configuration path in `opencode.json` / `opencode.jsonc` for OpenCode v2, replacing the v1 `plugin` / `provider` / `options` shape.

Adds `models.includeBy` and `models.excludeBy` filter arrays under `modelsDiscovery`, accepting `field` and `match` (regex) predicates to include or exclude discovered model IDs.

Adds `smartModelName` boolean setting under `modelsDiscovery` to derive display names from provider metadata rather than raw model IDs.

Adds `cache.ttlSeconds` setting under `modelsDiscovery.cache` controlling host-storage cache lifetime (defaults to 86400 seconds when `ttlSeconds` is omitted and cache is enabled).

Adds `filterNonChat` setting under `modelsDiscovery` (defaults to `true`) to filter non-chat models when the selected metadata source supplies that information.

+8 more

└──▷ **BREAKING ON UPGRADE**

- ! V2 does not read the V1 `auth.json` or `OPENCODE_AUTH_CONTENT` fallback — credential data stored in those locations is ignored under OpenCode v2.
- ! V1 XDG cache files are not imported into V2 — V2 cache and override storage use separate host-managed namespaces and must be rebuilt from scratch.
- ! The V1 commands `/models-discovery:config` and `/models-discovery:migrate` are not available in OpenCode v2.
- ! The V1 `modelsDiscovery.cache` disk-cache format is not used by OpenCode v2 — V2 caching stores through the host-provided `ctx.storage` API.

WAS THIS USEFUL?  

SMG Project Shepherd Model Gateway

Feedback ⓘ

v1.13.0 NOTES ↗ RANK

Engine-agnostic LLM gateway in Rust with full OpenAI & Anthropic API compatibility. Features gRPC pipeline, KV cache routing, chat history, embeddings, WASM plugins, and MCP.

> Shepherd Model Gateway's v1.13.0 release introduces the Symphony parser for reasoning and tool calls, overhauls KV-cache-aware routing with engine-side media preprocessing, and ships a Rust gRPC servicer path, alongside several breaking changes to CORS defaults, media host handling, schema migrations, Helm deployment, and metrics/error behavior.

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

01 **Symphony parser for reasoning and tool calls** NEW 95

Adds a `--parser symphony` flag (default remains `legacy`) and `X-SMG-Parser: symphony` request header to enable Symphony, a table-driven parser for reasoning and tool calls, as an opt-in beta; the response header `x-smg-parser` names the parser that handled the response. Publishes `smg-symphony` 0.1.0 and `smg-response-template` 0.1.0 on crates.io, and adds support for twelve new tool-call/reasoning formats: GLM 4.5 and 4.6, K2-Horizon, MiniMax M2 and M2.7, Inkling, MiniCPM5, dots3, Gemma 4, Mistral (Devstral 2, Ministral 3, Mistral Medium 3.5, Mistral Small 4), Llama Nemotron, FunctionGemma, bare JSON calls, and arguments-only forced calls. Also adds an `hf` parser name that compiles a checkpoint's `response_template` (transformers 5.18 format) into a Symphony table via the new `smg-response-template` crate.

— Flags, headers, crate names and twelve formats all spelled out v1.13.0

02 **KV-cache-aware routing with engine-side media preprocessing** NEW 90

Adds `--kv-index chain` to enable cache-aware routing fed by KV-cache events, and `--mm-processing worker` to move image/video preprocessing from the router to engine workers. This offload nearly doubles image request throughput (61.4 to 120.0 requests/s) and reduces median request time 3.3× on an 18-engine fleet.

— Two named flags with concrete throughput and latency numbers v1.13.0

03 **Rust gRPC servicer for vLLM, TokenSpeed, SGLang** NEW 85

Adds `--servicer-impl rust` switch to the Rust gRPC servicer, supported by vLLM (`vllm serve --servicer-impl rust`), TokenSpeed (`python -m smg_grpc_servicer.tokenspeed --servicer-impl rust`), and SGLang (`sglang serve --grpc-mode --servicer-impl rust`). `smg-grpc-servicer` 0.14.0 must be

installed alongside `smg` 1.13.0 on Rust servicer engine pods; the new servicer raises `TypeError` against the 1.12.0 `smg` wheel.

— Exact runnable commands for three engines plus version pairing requirement v1.13.0

04 **Private media host restrictions** BREAKING 70

Media URLs that resolve to private or loopback addresses are now refused. Adds `--mm-allow-private-media-hosts` gateway flag and `SMG_MM_ALLOW_PRIVATE_MEDIA_HOSTS=1` worker environment variable to restore the previous behaviour of permitting media fetched from private or loopback addresses.

— Flag and env var named with clear remediation v1.13.0

05 **Tokenizer L0 cache memory cap** IMPROVED 65

Adds `--tokenizer-cache-l0-max-memory` flag (default 256 MiB) to cap the tokenizer L0 cache byte budget, reducing live heap from ~930 MB to ~290 MB in steady-state runs.

— One flag with default and before/after memory numbers v1.13.0

06 **CORS closed by default** BREAKING 65

Adds `--cors-allowed-origins` flag to control which browser origins are allowed; CORS is now closed by default unless this flag is set. Browser clients relying on the prior open policy must pass `--cors-allowed-origins '*'` (or a specific origin list) on upgrade.

— Named flag with exact upgrade command but single surface v1.13.0

07 **Auto schema migration for history stores** BREAKING 65

Adds `auto_migrate: true` config key (or `DB_AUTO_MIGRATE=true` env var) in the store's schema config to apply schema migration 4 for Postgres and Oracle history stores. Operators must create the sweep index on existing tables before upgrading.

— Config key, env var and migration number named with upgrade note v1.13.0

08 **Helm chart readiness probe and security context changes** 65

BREAKING

The Helm readiness probe path moved from `/health` to `/readiness`, exposed via the new `router.readinessProbe.path` Helm key; deployments whose workers register only through the admin API must set `router.readinessProbe.path: /health`. Custom Helm router images must now run as UID 65532 or override `router.securityContext.runAsUser`.

— Two Helm config keys named with exact remediation values v1.13.0

09 **Faster tokenizer encode/decode via tokenizers-v1** IMPROVED

60

Adds opt-in `tokenizers-v1` build feature (off in published wheels and images) for 6.6–11.1× faster encodes of 80k-token prompts and ~17× faster full byte-level decode on MiniMax M3, GLM-5.3, Qwen3-8B and DeepSeek-V3.2.

— Named build feature and models but requires custom build to use `v1.13.0`

THINNER COVERAGE BELOW

10 **Constrained decoding for auto tool choice** NEW

55

Adds `constrained_decoding: always` worker config field so that `tool_choice: 'auto'` requests with tools are always grammar-constrained.

— Named config field but behaviour described briefly `v1.13.0`

11 **Metrics, error handling, and startup behavior changes** BREAKING

50

Python Router(...) metrics now bind all interfaces when `prometheus_host` is unset (v1.12.0 bound `127.0.0.1` only). Two metrics became histograms, error labels and error envelopes changed, and some requests that previously passed are now refused with `400`. An unreachable Postgres or Redis store, a mesh that cannot start, or malformed settings now prevents the gateway from binding a port at all, rather than failing at request time.

— Several named behavior changes but no single remediation path `v1.13.0`

└──▷ BREAKING ON UPGRADE

! CORS is closed by default; browser clients that relied on v1.12.0's open policy must pass `--cors-allowed-origins '*'` (or a specific origin list) on upgrade.

! Media URLs that resolve to private or loopback addresses are refused; set `--mm-allow-private-media-hosts` on the gateway and `SMG_MM_ALLOW_PRIVATE_MEDIA_HOSTS=1` on workers to restore the previous behaviour.

! Postgres and Oracle history stores require schema migration 4; set `auto_migrate: true` in the store's schema config or `DB_AUTO_MIGRATE=true`, and create the sweep index on existing tables before upgrading.

! The Helm readiness probe path moved from `/health` to `/readiness`; deployments whose workers register only through the admin API must set `router.readinessProbe.path: /health` or the router will never become Ready.

! Custom Helm router images must run as UID 65532 or override `router.securityContext.runAsUser`.

! The `smg-grpc-servicer` 0.14.0 must be installed together with `smg` 1.13.0 on Rust servicer engine pods; the new servicer raises `TypeError` against the 1.12.0 `smg` wheel.

! Python Router(...) metrics now bind all interfaces when `prometheus_host` is unset; v1.12.0 bound `127.0.0.1` only.

! Two metrics became histograms, error labels and error envelopes changed, and some requests that previously passed are now refused with 400; dashboards and API clients must be updated accordingly.

! An unreachable Postgres or Redis store, a mesh that cannot start, or a list of malformed settings now prevents the gateway from binding a port at all, rather than failing at request time.

WAS THIS USEFUL? 👍 👎

◆ Inference Serving & Optimization

Niko1221 Strata

Feedback ⓘ v0.2.5 NOTES ↗ RANK

Qwen3.8-Flash-Next on any consumer hardware: one-click install for Windows / Linux. Strata inference engine, OpenAI/Anthropic API on localhost, optional image input.

> Strata's biggest window yet: full DeepSeek V4.1-Flash support with a self-updating engine, a complete desktop GUI and Terminal CLI with signed engine updates, plus image input, parallel-request serving, and a long tail of per-architecture performance tuning across NVIDIA, AMD, and Intel hardware.

└──▷ WHAT SHIPPED · 20 FEATURES most completely described first ? what's the number?

01 DeepSeek V4.1-Flash model support and engine

NEW SUPPORTED PLATFORM NEW 96

Adds DeepSeek V4.1-Flash via `strata launch` and `strata setup` for PCs with a 12GB+ GPU and 46–64GB RAM, in `IQ1_M` (239GB), `IQ2_XXS` (257GB), and `IQ2_XS` (274GB) quantisation sets, shipped as a separate `strata-dsv41` engine package that coexists with the Qwen engine and self-updates on the next `strata launch` from a signed feed (an engine supplied via `--dsv41-package` or `STRATA_DSV41_EXE` is never replaced). Adds `--dspark on` / `--dspark auto` speculative-decoding flags raising throughput from ~31 to 55–69 tokens/s on 4x Radeon AI PRO R9700, `strata launch --model-path` for on-disk models including Unsloth UD-Q8_K_XL, a self-sizing RAM expert store (its 48GB cap removed in v0.2.5) with hot-expert profiles shipped with the engine, and a 32,768-token default thinking budget (`0` removes the limit). Adds `--speed` , `--full-quality` , and `--experts N` Terminal flags defaulting to Faster mode (5 of 6 experts) on PCs with 16GB VRAM or less and on Strix Halo systems (requires DeepSeek engine 0.1.50+), plus an in-app model picker entry with three size options, fixed 8K context, and no calibration step.

— Full mechanism, flags, quant sets and commands all named. v0.2.5 v0.2.4

03 Prompt chunk size control via --prefill

BREAKING 82

Adds a `--prefill` flag to control prompt chunk size; the default is raised to 4096/2048/1024 tokens (whatever fits), with `--prefill 512` restoring the prior behaviour. The larger default produces slightly different text on long prompts (mean KL 0.004); prompts shorter than the chunk are unchanged. Improves decode throughput for Qwen3.6-35B-A3B by up to 1.42x and prompt throughput by up to 1.80x on R9700, and improves long-prompt performance on AMD GPUs (131K tokens reads 8.7% faster, 230K 16% faster, 500K 79% faster on R9700).

— Flag, default values, and before/after behaviour named. v0.1.45

04 **NVIDIA/Intel kernel and path tuning env vars**

NEW SUPPORTED PLATFORM NEW 80

Adds `STRATA_DROP_CACHE_ON_EXIT=1` causing `strata serve` to release model-file page cache on exit (benefits dual-socket/multi-node Linux hosts); `STRATA_ROUTE_RESIDENT` opt-in running one warp per token on NVIDIA and AMD (+2–8% decode); `STRATA_TAIL_SKIP_MAX_CACHED=<percent>` to configure the VRAM occupancy threshold at which route tail skip turns off; an Intel Arc Pro B70 copy-engine prompt path (`STRATA_COPY_ENGINE=0` to disable) with a new expert-ranking profile at `data/expert-profile-sycl.bin` , raising expert cache hit rate from 60% to 71% and roughly halving prompt latency; adaptive-tier overlap of copies with the next decode window on Linux NVIDIA (`STRATA_ADAPT_OVERLAP=0` to disable; opt-in on Windows with `=1`); a dedicated kernel table for Tesla P100 (`STRATA_SM60_TABLE=0` to disable); interleaved `Q3_K` kernels for RTX 30-series `Q2_0` packs on Linux (`STRATA_MMVQ_IL_Q3K=0` to disable); measured PCIe share for canonical `Q2_0` packs on Linux (`STRATA_PCIE_AUTO_CANON=1` to opt in on Windows); and a NaN-guard for decode windows with all-NaN logits (`STRATA_NAN_GUARD=0` to disable), retrying from token 0 and logging non-finite logits (#879).

— Ten named switches each with a concrete behaviour or gain. v0.1.43

05 **Vision/image input support for Qwen** NEW 78

Enables pictures by default on NVIDIA and AMD cards: runs the vision reader on CPU for cards under 24GB and on GPU from 24GB (`--vision gpu` / `--vision cpu` / off still available per model); AMD now includes a GPU reader with automatic CPU fallback. Adds image input for Qwen3.6-35B-A3B via a `--vision` flag, downloading the pinned `mmproj` projector (0.9GB) and prompting 'Do you want images?' during configuration, plus a CPU image encoder path for NVIDIA cards under 24GB processing images in 4–6s at up to 300 tokens, with `--vision-tokens 1024` for higher detail.

— Flags and thresholds named across both releases. v0.1.45 v0.2.0-beta

06 **Parallel request serving for Qwen** NEW 78

Adds `--parallel N` (up to 8) and a `'parallel': N` config key to serve multiple simultaneous requests, yielding 1.14x–1.61x throughput over llama.cpp at equivalent slot counts, plus a `--batch-mtp` opt-in flag enabling draft-token verification across parallel slots (1.6x on one request, 1.2x on two).

— Flags, config key and concrete throughput numbers given. v0.1.45

07 **GPU-specific opt-in performance flags** NEW 78

Adds CUDA build option `-DSTRATA_MMQ_TAB=ON` with `STRATA_MMQ_DIRECT=1` and `STRATA_MMQ_GROUP_SORT=1` for pointer-table MMQ keeping resident experts in place (costs 36–38MiB per GPU; off in release builds); opt-in `STRATA_HIP_WMMA=1` for AMD GPUs (+44% prompt, +3.7% decode on R9700); opt-in `STRATA_EXCHANGE_ROTATE=1`

(+5% decode on RTX 3060 with resident RAM copy); opt-in `STRATA_PCIE_BALANCE=1` (+5.6% on dual A4000); and `STRATA_SM75_PERF=1` for Turing (sm_75) kernel series.

— Each flag named with a concrete measured gain. v0.1.45

08 **Built-in benchmark suite** NEW 78

Adds `BENCH.bat` / `./bench.sh` (`python tools/strata_bench.py`) benchmark suite that runs short chat, 4K and 16K prompts, and a five-turn agent session against a local install, with `--clients N` , `--long` , and `--compare` options, writing results in the community-benchmark format.

— Runnable commands and flags with described test scope. v0.1.43

09 **CUDA route tail-skip and PCIe auto-calibration defaults** 78

BREAKING

Enables `STRATA_ROUTE_TAIL_SKIP=7` by default on CUDA (one GPU or layer split): missed experts ranked 7th or lower across every token in a verify window are skipped entirely, yielding 13–16% faster decode on tested NVIDIA cards (disable with `STRATA_ROUTE_TAIL_SKIP=0`). Adds automatic PCIe share calibration for CUDA single-GPU setups: without `--pcie-frac` , the engine times the CPU pool in the first ~10s of decode and adjusts the share in steps of 0.05 between 0.10 and 0.60 (up to three adjustments), printing a `PCIe share:` line each time; restore the old rule with `STRATA_PCIE_FRAC_DEFAULT=old` . Both changes slightly alter generated answers compared with v0.1.41.

— Mechanism, thresholds and rollback flags all named. v0.1.42

11 **gfx12 AMD prompt-path defaults** IMPROVED 74

Enables five prompt-path switches (`STRATA_GDN_HEAD` , `GDN_PP=2` , `GDN_CONVL2` , `GDN_NOY` , `CVEC_FUSE`) by default for gfx1200/gfx1201 cards (e.g. Radeon AI PRO R9700), producing byte-identical residuals and logits with up to +1.8% prompt throughput at 32K tokens (disable with `STRATA_GFX12_DEFAULTS=0`); adds opt-in fused int8 prompt experts for gfx12 AMD cards via `STRATA_PF_FUSED=1` , delivering +15–19% prompt throughput at 4K–32K tokens on R9700 `IQ3_S` with lower first-token KL than the default path.

— All switches named with measured throughput changes. v0.1.42

12 **Adaptive tier and pinned-memory tuning env vars** NEW 72

Adds `STRATA_ROUTE_TAIL_SKIP=0` written by setup where every expert is in RAM (`STRATA_ROUTE_TAIL_SKIP=7` opt-in turns tail skip on for NVFP4 experts); `--adapt-async 1` to run the adaptive tier's expert swaps on a job thread instead of blocking decode (arena mode, one GPU, opt-in); `STRATA_PIN_BY_NEED=1` (Windows/WDDM, CUDA and HIP) to register experts the GPU will not hold as pinned memory in expert-

profile order, with `STRATA_ARENA_PIN_GIB` for the pin budget and `STRATA_PCIE_MIN1=1` as a separate switch; `STRATA_COMMIT_BATCH` (on by default; `=0` reverts to per-layer loop) and `STRATA_MTP_HIP_PRIO=0` to disable draft-stream high priority on gfx1200/gfx1201; and `STRATA_START_RETRIES` to retry a server start that dies inside the GPU runtime's own setup, at 2, 4, and 8 seconds.

— Every variable and its exact behaviour named. v0.1.44

13 **Quantization setup env vars for Unsloth files** NEW 70

Adds `STRATA_MMQ_KQ=1` to enable K-quant prompt kernels, written automatically into the run config's `env` block by setup for UD-Q5_K_XL and UD-Q6_K_XL files (prompts x1.3–1.7 on AMD with every expert in RAM), and `STRATA_HC_Q8=1`, written by setup for Unsloth UD-* files on NVIDIA and on by default for RDNA4 (gfx1200/gfx1201), reading hyper-connection projections from the file's own `Q8_0` bytes instead of a BF16-rounded copy.

— Named env vars and files, mechanism of the Q8 change given. v0.1.44

14 **AMD Radeon performance defaults** IMPROVED 68

Enables `--pipeline-windows 2` by default for gfx12 layer splits across 3 or more Radeon cards in `--serve` (disable with `--pipeline-windows 0` or `STRATA_PIPELINE_AUTO=0`); fused prompt kernels and 16K Linux prompt chunking by default on RDNA4 AMD Radeon (gfx1200/gfx1201) (disable with `STRATA_PF_FUSED=0` and `STRATA_GFX12_CHUNK=0`); and route tail skip and measured PCIe share defaults on AMD RDNA4 (disable with `STRATA_ROUTE_TAIL_SKIP=0` and `STRATA_PCIE_FRAC_DEFAULT=old`), delivering up to 38% faster decode on a 16GB Radeon.

— Named flags and measured gain, though defaults apply automatically. v0.1.43

16 **Hardware-specific inference kernel optimizations**

UPDATED SUPPORTED PLATFORM IMPROVED 64

Adds AVX-512 gather-based reading of i-quant tables on Intel Sapphire Rapids and newer server CPUs (25–47% faster expert-row reads), prompt kernel support for AMD gfx1151 matrix units on Strix Halo (22% faster prompts at 4K, 29% at 8K), new RDNA4 matrix-core kernels for Radeon AI PRO R9700 enabled by default (36% faster prompt throughput at 8K, 49% at 32K on 4 cards), and new AVX-512 (Zen 4/5) and AVX2 CPU row kernels for expert computation selected automatically (~25% faster decode, ~26% faster prompt reading on tested Zen 4 hardware).

— Named architectures and gains, but changes apply automatically. v0.2.5 v0.2.4

17 **Native Mac (Apple Silicon) engine support** NEW SUPPORTED PLATFORM

NEW 64

Strata now runs on M1 to M5 Macs (macOS 15 or newer, Command Line Tools, no Xcode) via `./setup.sh`, which builds the Metal engine from source; details in `docs/MAC.md`.

— Exact command and doc path given, limited mechanism depth. v0.1.44

---

18 **Single-file installers with on-demand engine download** BREAKING

62

Ships a single installer per platform — `Strata-0.2.5-windows-x64-setup.exe`, `Strata-0.2.5-linux-x86_64.AppImage`, and `Strata-0.2.5-macos-arm64.dmg` — where Windows and Linux installers no longer bundle an engine, instead downloading it on first launch with progress and resume support, or via the I already have the file offline path.

— Named installer files but limited behavioural detail. v0.2.5

---

19 **Multi-user server tuning environment variables** NEW

62

Adds multi-user server opt-in environment switches: `STRATA_SLOT_COPY_DIRECT=1`, `STRATA_SLOT_COPY_BATCHSYNC=1`, `STRATA_CKPT_POOL=1`, `STRATA_CKPT_THREADS=1`, `STRATA_ADMIT_MERGE_FIRST=1`, `STRATA_PIPE_BYIELD_NODRAIN=1`, `STRATA_BATCH_SELF_COMMIT=1`, `STRATA_QSA_SLOTS=1`, `STRATA_HANDOFF_PIPE=P`, `STRATA_ADMIT_SHORT_FIRST=<tokens>`, and `STRATA_TRACE` for admission-phase tracing.

— Every variable named but no individual mechanism given. v0.1.45

THINNER COVERAGE BELOW

---

22 **Image generation preview endpoint** NEW

54

Adds image generation in preview via `POST /v1/images/generations` (off unless a config asks for it; not yet offered in setup).

— Endpoint named but marked preview-only with no setup path. v0.1.44

---

23 **Qwen prompt-path and async-tier improvements** IMPROVED

44

Adds prompt-path selection based on prompt length (96 tokens and above use the chunked pass), eliminating timing-based path guessing that previously changed answers between runs, and background warm-up of the chunked pass after the model is ready, eliminating the one-time setup cost (~3.6s on an R9700) on the first prompt. Adds async-tier self-measurement for Qwen on single CUDA GPUs with all experts in RAM: the engine benchmarks for 64 windows and keeps the async path only if it is no more than 2% slower.

— Explains mechanism but no flag for the reader to use. v0.2.5

---

24 **Quantization quality comparison tool** NEW 43

Adds `tools/quant_quality.py` to rerun quantization quality comparisons against `Q8_0` on user-supplied files.

— Tool path named but no further detail on output or usage. v0.1.44

25 **Intel Arc Linux SYCL OS requirement bump** BREAKING 35

Intel Arc on Linux (SYCL) now requires Ubuntu 24.04 or newer (glibc 2.38).

— States the requirement but no migration guidance. v0.2.5

NEW SUPPORTED PLATFORM Docker image support for CUDA and ROCm flavors

└──▷ BREAKING ON UPGRADE

! Intel Arc on Linux (SYCL) now requires Ubuntu 24.04 or newer (glibc 2.38).

! The `--speed` / `--full-quality` / `--experts N` flags and the Speed step in the app require DeepSeek engine 0.1.50 or newer; an older engine receives no Speed step and ignores those flags.

! Windows and Linux installers no longer bundle a GPU engine; the engine must be downloaded on first launch or supplied via the I already have the file offline path.

! The larger default prompt chunk (4096/2048/1024 tokens) produces slightly different text on long prompts compared with the prior chunk-512 default (mean KL 0.004); use `--prefill 512` to restore the old behaviour. Prompts shorter than the chunk are unchanged.

! `STRATA_ROUTE_TAIL_SKIP=7` is now on by default for CUDA single-GPU and layer-split setups, which slightly changes generated answers compared to v0.1.41; set `STRATA_ROUTE_TAIL_SKIP=0` to restore previous answers.

! The PCIe share is now set automatically from a runtime CPU measurement rather than the start-up link probe alone, which changes decode behaviour (and answers via the `pcie_frac` path) for CUDA single-GPU setups; set `STRATA_PCIE_FRAC_DEFAULT=old` or pass `--pcie-frac N` to restore v0.1.41 behaviour.

v0.2.3 NOTES ↗ STABLE

v0.2.3 adds DFlash2 drafting opt-in, OS-level HTTPS certificate trust, agent chat context-window management, smarter CPU/VRAM defaults, and Terminal stability hardening.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.2.3 https://github.com/Niko1221/Strata.git
# already have the repo? check out this version:
$ git checkout v0.2.3
```

Adds `strata launch --drafting dflash2` to use the DFlash2 drafter (opt-in, Terminal only); other choices are `auto`, `mtp`, and `off`.

HTTPS now trusts the operating system's certificate store, so setups behind a corporate or school proxy no longer fail certificate checks.

Agent chat gains context-window awareness: reads the prompt size the model measured, checks a message against the model's window before sending, summarises earlier messages in pieces that fit, and recovers from `context too large` with `Summarize and continue` or `Start a new chat` options; `/compact` is supported.

Improves CPU worker and VRAM defaults: the app's Work profile now uses 5 CPU workers and keeps 1536 MiB of video memory free; conversation cache is sized from free memory (40% above a safety floor, capped at 8 GiB) when not explicitly set.

Improves Terminal stability: a second `strata launch` with the same choices prints `Already serving`; `--restart` handles context changes on the same port; `strata status` and `strata logs` find a running launch without an address; `--new-key` controls API key rotation; a busy port reports 'Port 8190 is already in use by another program'.

v0.2.2 NOTES ↗ STABLE

Strata v0.2.2 adds per-card GPU roles, optional API keys, NUMA-aware expert pooling, automatic VRAM fit reduction, and a unified models folder across the app and Terminal.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.2.2 https://github.com/Niko1221/Strata.git
# already have the repo? check out this version:
$ git checkout v0.2.2
```

Adds `--gpu-roles` flag to `strata launch` and `GPU_ROLES` Docker variable to assign each graphics card a Main, Helper, or Off role.

Adds `--no-api-key` flag and `"api_key": "none"` server config option to allow keyless server operation with a warning instead of a refusal.

Adds per-NUMA-node CPU expert pool splitting on Linux two-node machines with `STRATA_NUMA_POOL=0` to opt out; startup log reports the active path.

Adds automatic settings reduction (32K, 16K, 8K context, smaller prefill and reserve) when a model is slightly too large for available VRAM.

Adds unified models folder tracking between the app and the Terminal so both always share the same folder.

+3 more

└──▷ **BREAKING ON UPGRADE**

! Two equal GPU cards now put the lower-numbered card on Main, reversing the engine order from [0, 1] to [1, 0].

v0.2.1 NOTES ↗ STABLE

Strata v0.2.1 adds a task board UI, `strata bench run` benchmarking, `strata launch --mtp on|off`, `strata monitor`, faster parallel downloads, `user_args` for persistent engine settings, and new environment variables `STRATA_IGPU_PROMPT_SET` and `STRATA_HIPBLASLT_TABLE`.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.2.1 https://github.com/Niko1221/Strata.git
# already have the repo? check out this version:
$ git checkout v0.2.1
```

Introduces a task board view on the Tasks screen via a Schedule | Board switch, showing running, approval-pending, and failed tasks across all chats.

Adds `strata bench run` with suite `standard-v1`, a scorer, validator, and privacy-safe report; protocol in `docs/BENCHMARK_PROTOCOL.md`.

Adds `strata launch --mtp on|off` and Settings > Expert toggle to control the draft layer (MTP) per model.

Adds `strata monitor` to locate a running `strata launch` without `--url`.

Adds `STRATA_IGPU_PROMPT_SET` and `STRATA_HIPBLASLT_TABLE` environment variables for per-card tuning control.

+3 more

WAS THIS USEFUL? 👍 👎

BentoML

Feedback ⓘ v1.4.40 NOTES ↗ RANK

BentoML packages models and AI applications into inference APIs, job queues, and multi-model pipelines, and deploys them to Kubernetes or BentoCloud.

> BentoML added agent skills for scaffolding and deploying projects, and tightened dependency pinning (uv, pip, certifi) in generated Dockerfiles, including a breaking certifi bump.

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

01 **Pinned uv and pip versions in generated Dockerfiles** IMPROVED

70

Generated Dockerfiles and local `bentoml build` now install `uv==0.13.4` with `--require-hashes` when uv is missing or older than 0.13.4, and use pip 26.2.2 where supported.

– Exact versions and flag named, command context given v1.4.40

THINNER COVERAGE BELOW

02 **Required certifi version bump** BREAKING

50

`certifi>=2026.10.19` is now required; projects pinned to an older certifi version will need to upgrade.

– Exact version named with clear migration note v1.4.40

03 **Agent skills for project creation and deployment** NEW

35

Adds agent skills for creating BentoML projects and deploying to Kubernetes and EC2.

– Names Kubernetes and EC2 but no commands or mechanism given v1.4.40

└──▷ **BREAKING ON UPGRADE**

! `certifi>=2026.10.19` is now required; projects pinned to an older certifi version will need to upgrade.

WAS THIS USEFUL? 👍 👎

vLLM

Feedback ⓘ announced · Oct 23, 2026 BLOG ↗

The vLLM inference engine runs large language models with high-throughput, memory-efficient GPU execution.

vLLM adds day-0 `vllm serve` flags for MiniMax-M3.1-Flash: NVFP4 KV cache, DSpark speculative decoding, indexer DCP, MXFP8 MoE, and prebuilt Docker images for NVIDIA and AMD.

NEW SUPPORTED PLATFORM Adds `--kv-cache-dtype nvfp4` and `--attention-config '{"indexer_kv_dtype": "nvfp4"}'` support for MiniMax-M3.1-Flash NVFP4 KV cache serving.

Adds `--reasoning-parser minimax_m3` to `vllm serve` for MiniMax-M3.1-Flash reasoning output parsing.

Adds `--speculative-config` with `method: dspark`, `attention_backend: FLASHINFER_TS` or `TRITON_ATTN`, and `draft_sample_method: probabilistic` for DSpark speculative decoding.

Adds `--attention-config '{"indexer_decode_cp": true}'` for indexer decode context parallelism, up to 3.3× faster per sparse layer at 512K context on 4× GB300.

NEW SUPPORTED PLATFORM Adds `--quantization-config.moe mxfp8` for MXFP8 MoE quantization on AMD MI350-series GPUs.

+3 more

WAS THIS USEFUL? 👍 👎

NVIDIA Dynamo

Feedback ⓘ v1.6.0 NOTES ↗ RANK

Dynamo is a datacenter-scale distributed inference serving framework with disaggregated prefill and decode, KV-cache-aware routing, and multi-engine support.

> Dynamo v1.6.0 consolidates router configuration into a single pluggable policy file with a classifier plugin hook, adds OpenAI-compatible reranking and runtime endpoint-floor APIs, and extends observability and transport with OTLP tracing, a QUIC response plane, and a gRPC KV-pool relay.

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

01 thinking_token_budget field for reasoning caps NEW

80

A root-level `thinking_token_budget` field caps reasoning tokens on SGLang and vLLM backends, settable per request through the chat completions API.

Cap reasoning token spend on a per-request basis when calling a vLLM or SGLang model through the Frontend.

```
$ curl -X POST http://localhost:8000/v1/chat/completions \
  -H 'Content-Type: application/json' \
  -d '{"model": "deepseek-r1", "thinking_token_budget": 512,
"messages": [{"role": "user", "content": "Explain KV
caching."}]}'
```

— Field named with a runnable curl example v1.6.0

02 Unified router policy configuration file BREAKING

70

Router scoring knobs are consolidated into a single `--router-policy-config` flag, deprecating the older individual scoring flags; existing deployments using those flags must migrate their configuration to the new file.

Switch a deployment to the new unified router policy file instead of individual scoring flags.

```
$ dynamo serve --router-policy-config /etc/dynamo/router-
policy.yaml
```

— Names the flag and gives a runnable migration command v1.6.0

03 DYN_PARSER_VERSION opt-in V2 parsers NEW

65

The `DYN_PARSER_VERSION` environment variable selects opt-in V2 tool and reasoning parsers for eight model families.

— Names the env var but gives no exact values or example v1.6.0

04 **/v1/min-endpoints API for Dynamo Planner** NEW 65

Adds a `/v1/min-endpoints` API endpoint so Dynamo Planner can reconfigure endpoint floors and GPU budget bounds at runtime.

— Endpoint and purpose named but no example request v1.6.0

05 **Request-classifier plugin hook in Dynamo Router** NEW 60

A new plugin hook lets a request classifier defer a request, override scheduling inputs, or suggest a worker without overriding explicit worker pins; Dynamo ships ThunderAgent as a builtin router plugin using this hook.

— Mechanism described but no config key or command shown v1.6.0

06 **OpenAI-compatible /v1/rerank endpoint** NEW 60

Adds an OpenAI-compatible `/v1/rerank` endpoint serving SGLang cross-encoder models.

— Endpoint named but no example call or parameters v1.6.0

THINNER COVERAGE BELOW

07 **OTLP metrics export and per-request tracing** NEW 55

Adds OTLP export of Dynamo and engine metrics to an OpenTelemetry Collector, plus opt-in per-request lifecycle tracing from arrival to completion.

— Describes behaviour but no config key or setup step v1.6.0

08 **QUIC response plane** NEW 40

Adds an opt-in batched QUIC response plane alongside the default TCP path.

— States the addition with no mechanism or enabling steps v1.6.0

09 **Session prefix index for agentic workloads** NEW 40

Adds an opt-in session prefix index tracking session-to-block lineage for agentic workloads.

— Thin description, no config or usage shown v1.6.0

10 **gRPC relay for KV pool state** NEW 35

Exposes a versioned gRPC relay for KV pool state to consumers outside the data center.

— Names the surface but gives no endpoint details or usage v1.6.0

└──▷ BREAKING ON UPGRADE

! Router scoring flags are deprecated in favour of `--router-policy-config`; existing deployments using the older individual flags must migrate their configuration.

WAS THIS USEFUL? 👍 👎

vla.cpp

Feedback ⓘ v0.4.0 NOTES ↗ RANK

A unified inference runtime for VLA models.

> vla.cpp v0.4.0 adds OpenVINO and Snapdragon (Hexagon/Adreno) backends, a native `--text` inference path that skips the Python tokenizer, two new models (Octo-Small and TurboVLA), and a wider set of prebuilt binaries alongside several breaking packaging and config changes.

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

01 Native --text inference without Python tokenizer NEW

85

The `vla-cli` command gains a `--text` flag that reads the tokenizer embedded in the GGUF file, letting prompts run natively without any Python tokenizer dependency; this is now supported for Octo, π0, π0.5 and OpenVLA-OFT models.

Run a prompt natively on Octo-Small without any Python tokenizer dependency, using the embedded GGUF tokenizer.

```
$ ./build/vla-cli -hf vrfai/octo-small-libero-gguf --image
assets/front.jpg --text "pick up the black bowl" --pretty
```

— Flag, command, and runnable example all given. v0.4.0

02 Prebuilt binaries and packaging changes BREAKING

75

Ships prebuilt binaries for `linux-x86_64-cpu`, `linux-x86_64-cuda-12.8`, `linux-x86_64-cuda-13.4`, `linux-aarch64-cpu`, `linux-aarch64-cuda-13.4`, and `macos-arm64-metal`, plus a Python wheel. The x86 CUDA 12.8 tarball is renamed to `linux-x86_64-cuda-12.8`, and the macOS tarball now ships only `vla-cli` and `vla-bench`.

— All tarball names and rename called out. v0.4.0

03 Build option rename and config file enforcement BREAKING

70

The CMake option `VLA_OCTO` is renamed to `VLA_SPM` (the old name still works). Command-line flags now override `--config`, and a missing or invalid config file is now a hard error rather than silently ignored.

— Exact flag and option names with before/after behavior. v0.4.0

04 OpenVINO backend for Intel hardware NEW

65

Adds an OpenVINO backend targeting Intel CPUs, iGPUs and NPUs, delivering 3-10x faster inference than the CPU backend on an Arc iGPU; see `docs/backend/ov.md` for setup.

— Named backend and docs path but no exact command. v0.4.0

---

05 **Octo-Small model support** NEW 65

Adds the Octo-Small model (`vrfai/octo-small-libero-gguf`) with a ggml inference engine supporting diffusion, L1/proprio action heads and the LIBERO client.

— Names model id and supported engine components. v0.4.0

---

06 **Snapdragon X Hexagon/Adreno backend support** NEW 60

Adds Snapdragon X on Windows on Arm support with Hexagon NPU and Adreno GPU backends and automatic CPU fallback; see `docs/backend/hexagon-windows.md` .

— Backend named with docs path, no direct command. v0.4.0

THINNER COVERAGE BELOW

---

07 **TurboVLA model support** NEW 35

Adds the TurboVLA model (`vrfai/turbovla-libero-gguf`).

— Only the model name is given, no further detail. v0.4.0

---

08 **Faster CUDA inference** IMPROVED 35

Delivers 6-18% faster inference on CUDA with identical outputs compared to the previous release.

— Numeric improvement given but no mechanism or command. v0.4.0

└──▷ BREAKING ON UPGRADE

! The x86 CUDA 12.8 tarball is renamed `linux-x86_64-cuda-12.8` ; the macOS tarball ships `vla-cli` and `vla-bench` only.

! The CMake option `VLA_OCTO` is renamed `VLA_SPM` (the old name still works).

! Command-line flags now override `--config` , and a missing or invalid config file is now a hard error rather than silently ignored.

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

Open WebUI

Feedback ⓘ v0.12.4 NOTES ↗ RANK

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 v0.12.4 adds an `ENABLE_SKILL_LAZY_LOADING` option to send skills to models only on demand, a `UI_VISIBILITY` admin control to hide individual chat controls from non-admin users, and extends context compaction to temporary and short chats, alongside several smaller UI and permission refinements.

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

01 UI_VISIBILITY admin control for chat interface elements NEW

90

Adds `UI_VISIBILITY` setting (also configurable under Admin Settings > Interface > Visible controls) letting admins hide individual controls — including model capabilities, Capture, Compare text, Attach Webpage, Attach Files, Attach Notes, Attach Knowledge, Reference Chats, `/status`, `/model`, `/settings` commands, Fork, Merge Responses, Preview, Artifacts, Archive, Tags, and more — from non-admin users without disabling the underlying features.

Simplify the chat UI for end users by hiding controls they cannot or should not use, such as Fork, Artifacts, and the `/settings` command.

📍 In the Admin panel, go to Admin Settings › Interface › Visible controls and deselect Fork, Artifacts, and `/settings`, or configure `UI_VISIBILITY` accordingly in your environment.

— Named config key, admin path, and full list of controllable items. v0.12.4

02 Lazy skill loading via ENABLE_SKILL_LAZY_LOADING NEW

80

Adds `ENABLE_SKILL_LAZY_LOADING` environment variable (also toggled via Admin Settings > General > Load selected skills on demand) so skills attached to a model are only sent to it on demand rather than with every message.

Enable lazy skill loading so models only receive skill instructions when actually needed, reducing prompt overhead on every message.

📍 In the Admin panel, go to Admin Settings › General and enable Load selected skills on demand, or set the environment variable `ENABLE_SKILL_LAZY_LOADING=true` when starting Open WebUI.

— Names exact env var and admin toggle path; mechanism is clear. v0.12.4

03 **Context compaction extended to temporary and short chats**

IMPROVED 70

Extends context compaction to temporary chats and short chats already over the context threshold, preventing context-length errors on follow-up, regenerate, or edit in those sessions, and adds a Context compacted inline marker in the chat view showing where older messages were summarised, expandable to reveal the summary, visible in Chat Controls > Overview, with compaction progress and errors surfaced inline instead of as pop-up notices.

— Explains mechanism and scope; gives a UI path to inspect it. v0.12.4

04 **Automatic hiding of unavailable and unauthorized options**

IMPROVED 60

Hides unavailable options automatically (e.g. Capture without a Vision model, Attach Notes without the Notes permission, Context Compaction Threshold while compaction is off, Web Search while unavailable, `/skills:create` for incapable models) rather than showing them greyed out, and hides the Access button and menu entry for models, prompts, tools, knowledge, skills, notes, and folder settings from users who lack the matching sharing permission.

— Lists named controls but behavior is automatic, no user action. v0.12.4

THINNER COVERAGE BELOW

05 **Interactive profile cards for owner names** IMPROVED 50

Replaces email tooltips on owner names in workspace lists (models, knowledge, prompts, skills, tools) and admin Functions with interactive profile cards including profile pictures; profile pictures in access settings, version history, and Analytics also open the card.

— Names several surfaces where the card appears, no further mechanism. v0.12.4

06 **Start new chat directly from folder sidebar** NEW 40

Adds a New Chat item to a folder's sidebar menu to start a chat directly inside that folder.

— Clear UI entry point but thin on further detail. v0.12.4

07 **Turn count shown in chat sidebar tooltip** NEW 35

Shows the user and assistant turn count for the current branch in the sidebar chat-preview tooltip.

— Describes the feature but no interaction or configuration path. v0.12.4

08 **Per-setting saves in Chat Controls** IMPROVED 30

Saves each Chat Controls change (system prompt, parameters, etc.) individually per setting rather than as a whole snapshot.

— Describes behavior change but no mechanism or surface named. v0.12.4

09 **Exact timestamp on hover for last-active times** IMPROVED 30

Shows exact date and time on hover for the last-active timestamp in the admin user list and in a group's member list.

— Small UI detail with a clear location but no further mechanism. v0.12.4

WAS THIS USEFUL? 👍 👎

Ollama

Feedback ⓘ v0.40.11 NOTES ↗ CODE ↗ RANK

Ollama downloads, serves, and chats with local language models through a CLI and REST API.

> Ollama added support for NVIDIA's Nemotron 3.5 Bolt E4B, a Gemma 4-based vision-language model with a 128K-token context window and speculative decoding.

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

01 **Nemotron 3.5 Bolt E4B model support** NEW 88

Ollama added `nemotron-3.5-bolt:e4b`, NVIDIA's Gemma 4-based vision-language model, supporting image understanding, tool use, and a 128K-token context window, with a DFlash draft for speculative decoding. Run it with `ollama run nemotron-3.5-bolt:e4b`.

Run the Nemotron 3.5 Bolt E4B model locally for vision-language tasks or tool use with a 128K-token context window.

```
$ ollama run nemotron-3.5-bolt:e4b
```

— Named model, exact run command, and concrete capabilities given v0.40.10

NEW INTEGRATION OpenAI-compatible speech endpoint

NEW SUPPORTED PLATFORM d3 decision model family

v0.40.11 NOTES ↗ CODE ↗ LATEST STABLE

Adds on-device speech synthesis with voice catalogs, a streaming audio endpoint, and d3 decision models for classification and probability scoring.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.40.11 https://github.com/ollama/ollama.git
# already have the repo? check out this version:
$ git checkout v0.40.11
```

Streams synthesized speech through the OpenAI-compatible `/v1/audio/speech` endpoint, letting apps consume audio as it is generated.

Adds `ollama run` speech playback on macOS: runs a speech model (e.g. `phonon`) and plays its audio output aloud in real time.

> Adds `ollama show` voice listing: displays a `Voices` table when the model exposes a voice catalog.
>
> Adds `ollama create` voice import: reads `voices/NAME.safetensors` from the checkpoint folder and registers named voices for Phonon.
>
> Adds the `d3` model family (2B, 4B, 9B, 27B) for decision tasks — classification, yes/no answers, and probability scoring — via `ollama run d3` with the `--ask` flag.
>
> `+1 more`

WAS THIS USEFUL? 👍 👎

tensorsharp

Feedback ⓘ v2026.10.23 NOTES ↗ RANK

A native .NET LLM inference engine for GGUF models with console, web chatbot, iPhone app, and Ollama/OpenAI-compatible APIs. Supports GPU on Windows/MacOS/iOS/Linux.

> TensorSharp introduces personal knowledge collections with BM25 and optional semantic search, exposing new server-side APIs and a matching knowledge management experience in TensorAgent Desktop.

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

01 Personal knowledge collections with BM25/semantic search

NEW SUPPORTED PLATFORM NEW 95

Setting `TS_KNOWLEDGE_DIRECTORY` on `TensorSharp.Server.Host` enables a personal knowledge library backed by local BM25 search, with optional GGUF-embedding semantic search. New `/api/knowledge` REST routes manage knowledge sources: uploading originals, searching, reading, reindexing, and deleting. Up to 32 collection IDs can be selected via `knowledgeCollections` in `POST /api/chat` (WebUI), `POST /v1/chat/completions`, `POST /v1/responses`, `POST /api/chat/ollama`, and `POST /api/generate` to scope retrieval per inference request.

Enable the personal knowledge library on a running server so any inference request can retrieve from local documents.

```
$ TS_KNOWLEDGE_DIRECTORY=/data/knowledge dotnet run --project
TensorSharp.Server.Host -c Release -
p:TensorSharpSkipMlxNative=true -- --model models/gemma-4-E4B-
it-Q8_0.gguf --backend ggml_cuda --max-tokens 512
```

Scope a chat completion to specific knowledge collections so only those sources are retrieved as evidence.

```
$ curl -X POST http://localhost:5000/v1/chat/completions \
  -H 'Content-Type: application/json' \
  -d '{"model":"gemma-4-E4B-it-Q8_0","knowledgeCollections":
["col-001","col-002"],"messages":
[{"role":"user","content":"Summarise the Q3 findings from my
uploaded reports."}]}'
```

— Full mechanism, named endpoints and limits, with runnable examples v2026.10.11

THINNER COVERAGE BELOW

02 **Knowledge collections in TensorAgent Desktop**

NEW SUPPORTED PLATFORM NEW 55

TensorAgent Desktop gains knowledge collection support via Menu → Knowledge, enabling file uploads, recursive folder uploads, saved-chat snapshots, source previews, and per-chat collection selection.

— UI path given but no API, flag, or config detail v2026.10.11

NEW SUPPORTED PLATFORM DeepSeek V4.1 Flash 1017 with unsloth GGUFs, device-resident TP, YOCO on by default

UPDATED SUPPORTED PLATFORM MiniMax-H3 and LTX 2.5 exact load-path and graph speedups

NEW SUPPORTED PLATFORM Qwen Image 2.1 Turbo model support

NEW INTEGRATION Nomic Embed 1.5 support

NEW SUPPORTED PLATFORM Tensor parallelism for Qwen 3.8 and DeepSeek V4.1

NEW SUPPORTED PLATFORM Compact CUDA expert caching for Qwen3.8 host offload

NEW SUPPORTED PLATFORM Experimental CUDA 13 GB10 / DGX Spark support

NEW INTEGRATION Bonsai 2 model support

NEW INTEGRATION DiffusionGemma as Jev model

NEW INTEGRATION Unsloth Qwen-Image-2.1 GGUF model support

RETIRED INTEGRATION Legacy qwen-image-edit-2511 model removed

RETIRED INTEGRATION Bonsai 1 and qwen3 plug-in removed

v2026.10.23 NOTES ↗ LATEST STABLE

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2026.10.23
https://github.com/zhongkaifu/TensorSharp.git
# already have the repo? check out this version:
$ git checkout v2026.10.23
```

v2026.10.10 NOTES ↗ STABLE

TensorAgent UX updated in v2026.10.10.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2026.10.10
https://github.com/zhongkaifu/TensorSharp.git
# already have the repo? check out this version:
$ git checkout v2026.10.10
```

Updates the TensorAgent desktop UX.

v2026.10.08 NOTES ↗ STABLE

Edited images are now downloadable in the UI.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2026.10.08
https://github.com/zhongkaifu/TensorSharp.git
# already have the repo? check out this version:
$ git checkout v2026.10.08
```

Makes edited images downloadable in the UI.

v2026.10.03 NOTES ↗ STABLE

TensorAgent Desktop releases for macOS and Windows, gaining multi-language UI, per-turn stats, and Enter-key prompt send.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2026.10.03
https://github.com/zhongkaifu/TensorSharp.git
# already have the repo? check out this version:
$ git checkout v2026.10.03
```

Releases TensorAgent Desktop as a downloadable app for macOS (Apple Silicon, macOS 14+) via `tensoragent-desktop-2026.10.03-osx-arm64.dmg` and Windows x64 via `win-x64-cpu.msi` / `win-x64-cuda.msi`.

Adds multi-language support to TensorAgent, enabling non-English interface and conversation use.

Adds per-turn statistics display at the end of each chat turn in TensorAgent.

Adds Enter-key send in TensorAgent, allowing prompt submission without clicking a button.

v2026.09.29 NOTES ↗ STABLE

Adds multi-sub-agent delegation, Qwen-Image-2.1 LoRA, Radix cache for speculative decoding, new layer-split parameters, and image thumbnail UI.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2026.09.29
https://github.com/zhongkaifu/TensorSharp.git
# already have the repo? check out this version:
$ git checkout v2026.09.29
```

Enables Radix cache for speculative decoding, extending prefix-sharing to speculative decode workloads.

Adds support for running multiple sub-agents with bounded delegation.

Adds LoRA support for Qwen-Image-2.1.

Adds new parameters for layer split configuration.

Supports more data types in Jev mode.

+1 more

└──▷ BREAKING ON UPGRADE

! Bonsai 1 model and the qwen3 plug-in are removed; TensorAgent now offers Bonsai 2 27B instead.

! The legacy `qwen-image-edit-2511` model is removed.

WAS THIS USEFUL? 👍 👎

mlxtop

Feedback ⓘ v3.1.2 NOTES ↗ RANK

The mlxtop terminal UI monitors local LLMs on Apple Silicon, showing loaded models, memory use, GPU activity, and generation speed.

> mlxtop added AMD GPU monitoring support on Linux via sysfs, plus more accurate live generation-rate tracking and compatibility with newer mlx-community server token formats.

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

01 **AMD GPU support on Linux** NEW SUPPORTED PLATFORM NEW 75

mlxtop now reads AMD GPU stats via amdgpu sysfs files under `/sys/class/drm/card*/device`, with no ROCm install needed. Cards put to sleep by the kernel (`power/runtime_status` other than `active`) remain listed with VRAM but without utilization or temperature, avoiding spurious wake events on kernels before 6.13. New `[` / `]` keybindings on Linux select a GPU card, and new GPU findings for AMD cards cover placement and VRAM; clock-limit reasons are unavailable on AMD, so the STATE column shows load only.

— Names sysfs path, kernel setting, keybindings and STATE column behavior. product docs

THINNER COVERAGE BELOW

02 **Per-request generation rate calculation** UPDATED INTEGRATION IMPROVED 54

mlxtop computes per-request generation rate live from `generated_tokens` between samples, counting only when the same slot runs the same task in both samples, so finished or newly started requests never contribute a spurious speed reading.

— Mechanism explained but it's a passive accuracy fix, nothing to act on. product docs

03 **next_token parsing for newer server builds** IMPROVED 40

mlxtop now reads `next_token` both as an object and as the per-sequence list returned by newer mlx-community server builds, improving compatibility with updated server responses.

— Names the field but gives no further mechanism or user action. product docs

v3.1.2 NOTES ↗ LATEST STABLE

> Per-request token counts and speeds now appear in the Overview queue for oMLX, mlx-serve, and Splash; Prompt load panel gains a generated-tokens row.

└──▷ GET THIS VERSION ▾

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

Adds per-request generation speed to the Overview queue, showing each running request's token count and tokens/s (e.g. `8f31a2 1.7k · 41.0 tok/s`) for oMLX, mlx-serve, and Splash.

Shows each running request's generated token count in a dedicated row under the Prompt load panel's size labels when the panel is tall enough to display four rows.

WAS THIS USEFUL? 👍 👎

llamastash

Feedback ⓘ v0.9.0 NOTES ↗ RANK

Zero-overhead, terminal-native local-LLM orchestrator. Run any engine, any model via the same proxy.

> LlamaStash's three releases built out a full observability stack — the `llamastash requests` log and `GET /metrics` Prometheus endpoint — added a `session` subcommand to point any coding tool at a local model for one run, let engines report their own model catalogs via `rows_from`, and shipped a built-in chat UI, while removing native Lemonade backend support and reshuffling CLI output fields.

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

01 **`llamastash requests` log and CLI** BREAKING 95

The `llamastash requests` subcommand reads the proxy request log with `llamastash requests <model>`, `llamastash requests <launch-id>`, `-n <count>`, and `--json` flags, backed by `proxy.request_log_file: true` in `config.yaml` which persists to `<log dir>/requests.jsonl` across daemon restarts. Later releases added a `PP/S` column and `Agg tok/s` / `Agg PP tok/s` aggregate columns to the table and summary, `decoding_speed_tps` and `prefill_speed_tps` fields from engine-reported `usage` data, `alias` notes for requests routed via `proxy.aliases`, `ui` notes for browser-originated chat traffic, and a Shift+Q TUI keybind to open the log for the model under the cursor. Note: the `PP/S` column insertion shifted later columns (breaking `awk -F\t` scripts by position) and the summary was split into two rows with `Prefill tok/s` renamed to `Avg PP tok/s`.

Tail the last 20 requests to a specific model as JSON to feed into a script or SIEM pipeline.

```
$ llamastash requests qwen3.8-27b-q8_0 -n 20 --json
```

— Full flag/field history plus exact breaking-change mechanics given. v0.8.0 v0.9.0 v0.7.0

02 **`GET /metrics` Prometheus endpoint** NEW 95

A `GET /metrics` endpoint on the proxy port exposes Prometheus counters (`llamastash:requests_total`, `llamastash:request_errors_total`, `llamastash:prompt_tokens_total`, `llamastash:request_duration_seconds_*`, `llamastash:requests_in_flight`, `llamastash:request_timed_total`), filterable with `?model=`, `?format=json`, and `?launch=1`; a `GET /metrics/backend` endpoint merges every ready launch's own `/metrics` in parallel, labeled by `launch` and

`model`; and `--metrics` / `--enable-metrics` flags on `llamastash start` expose per-engine metrics for llama.cpp and SGLang. A later release added `llamastash:prefill_tokens_total`, `llamastash:prefill_seconds_total`, `llamastash:request_prefill_tokens_per_second{model}`, and `llamastash:request_tokens_per_second_aggregate{model}`.

Scrape all engine metrics in one shot — LlamaStash proxy counters plus every running backend — into a single Prometheus document for Grafana.

```
$ curl -s localhost:11435/metrics/backend
```

Pull aggregate and per-request throughput metrics — including the new prefill counters — from the proxy's Prometheus endpoint to compare concurrency impact.

```
$ curl -s http://127.0.0.1:11435/metrics | grep -E 'llamastash:
(prefill|request_tokens_per_second_aggregate)'
```

— Endpoints, query params and metric names all named with examples. v0.7.0 v0.8.0

03 **`llamastash session` subcommand** NEW 95

`llamastash session` launches `claude`, `codex`, `opencode`, `pi`, or `aider` against one proxy-served model for that process only, using env vars, `-c` argv overrides, or a generated extension file — never writing the tool's global config. `--dry-run` prints a shell-quoted launch line, and `--dry-run --json` prints the resolved launch as a JSON object.

Point Claude Code at a specific local model for one session without touching its global config — useful for a focused coding task on a large-context model.

```
$ llamastash session claude Qwen3.8-27B-UD-Q6_K
```

Preview the exact env-var launch line that `session` would use before actually starting anything — paste the output into a script or CI job.

```
$ llamastash session claude Qwen3.8-27B-UD-Q6_K --dry-run --json
```

— Exact subcommand, flags and runnable example commands given. v0.9.0

04 **`rows_from` engine-owned model catalogs** NEW 80

The `rows_from` key on `backend.generic.servers` entries in `config.yaml` lets an engine report its own installed model catalog via fields `command`, `items`, `name`,

`when`, `arch`, `params`, `quant`, `ctx`, and `memory_gib`, so no `proxy.aliases` lines are needed per model.

Let FastFlowLM advertise its own installed models to llamastash so each appears as a startable row without hand-coding aliases.

```yaml
backend:
  generic:
    servers:
      - name: flm
        binary: ~/.local/opt/fastflowlm/flm
        args: [serve, "{name}", --host, "{host}", -p, "{port}",
--cors, "0"]
        ready: /v1/version
        rows_from:
          command: [~/.local/opt/fastflowlm/flm, list, --filter,
installed, --json]
          items: models
          name: model
          when: installed
          arch: details.family
          params: details.parameter_size
          quant: details.quantization_level
          ctx: default_context_length
          memory_gib: footprint
```

— All config fields named with a worked FastFlowLM example. v0.9.0

---

05 **Memory gate accuracy for hybrid Qwen models** IMPROVED

80

Improves memory gate accuracy for hybrid Qwen models (Qwen3-Next, Qwen3.5, Qwen3.8-27B) by pricing KV cache only on attention layers and excluding MTP layers when MTP is off, reducing the estimated VRAM for Qwen3.8-27B `Q4_K_M` at 64K from ~26.5 GiB to ~20 GiB.

— Mechanism and concrete before/after VRAM numbers given. v0.7.0

---

06 **Lemonade backend removed, migrate to `backend.generic`** BREAKING

70

A `config.yaml` block with a `backend.lemonade:` key is now rejected with an error; the `--lemonade` CLI flag and `LLAMASTASH_LEMONADE` environment variable are removed. Migration path: use a `backend.generic` entry per RECIPES.md § Lemonade.

— Names removed flag/env var and points to migration doc. v0.9.0

---

07 **Built-in chat UI for engines without one** NEW 70

Serves the llama.cpp UI at `http://127.0.0.1:11435/ui/` for engines that have no native web UI (vLLM, SGLang, generic entries such as gufo or Halogen), with a model chooser at `/ui/switch` and session pinning via cookie.

— Named URL paths and pinning mechanism, but no example. v0.8.0

08 **`proxy.aliases` for arbitrary model name mapping** NEW 65

The `proxy.aliases` config key in `config.yaml` maps arbitrary model names (e.g. `gpt-4o-mini`, `claude-haiku`) to local models, surfacing the aliases in `/v1/models` and `/api/tags` so hard-coded clients can route to local models unmodified.

Let a hard-coded tool that only speaks `gpt-4o-mini` route to a local Qwen model without touching the tool's config.

yaml

```yaml
proxy:
  aliases:
    gpt-4o-mini: qwen3.8-27b-q8_0
    claude-haiku: unsloth/Demo-GGUF/demo-Q4_K_M
```

— Config key, endpoints and a runnable YAML example given. v0.7.0

THINNER COVERAGE BELOW

09 **`llamastash completions <shell>` subcommand** NEW 55

Prints a shell completion script for bash, zsh, fish, PowerShell, or Elvish with no daemon required.

— Clear command and supported shells, but no example shown. v0.7.0

10 **`RECIPES.md` tested configuration recipes** NEW 50

Adds `RECIPES.md` with tested `config.yaml` blocks, wrapper scripts, and knob notes for llama.cpp (hip/vulkan/rocm builds), vLLM, SGLang, gufo, Halogen, CIRU, ds4, Lemonade, FastFlowLM on AMD NPU, and whisper.cpp, plus model recipes for Qwen3.8 Flash-Next, Qwen3.8 27B, DeepSeek V4, GLM, Gemma 4 with MTP, and Qwen3.5 4B vision.

— Lists covered engines/models but no mechanism or command. v0.9.0

11 **`doctor --fix` and `--dry-run` repair flags** NEW 45

Adds `doctor --fix` flag to repair findings with safe mechanical fixes, and `--dry-run` to preview repairs without applying them.

— Flags named but no detail on what findings are fixed. v0.7.0

12 **`/api/show` launch-scoped addressing** IMPROVED 40

Extends `/api/show` to accept `<model>@<launch>` addressing, consistent with other chat surfaces.

— Endpoint and syntax named, but no example or rationale. v0.7.0

13 **`bell: false` config key** NEW 35

Adds a `bell: false` config key in `config.yaml` to suppress the terminal bell that rings when a pull or launch completes.

— One config key with clear effect, minimal surrounding context. v0.7.0

14 **`backend.generic` env `~/` path resolution** IMPROVED 30

`backend.generic` `env` values now resolve a `~/` `-prefix` against the home directory, matching how `binary` already resolves paths.

— Single-line change with no example or broader context. v0.9.0

15 **`status --json` drops `lifecycle` field** BREAKING 30

`status --json` backend rows no longer include the `lifecycle` field or the umbrella columns that accompanied it.

— Names the removed field but gives no migration guidance. v0.9.0

└──▷ BREAKING ON UPGRADE

! `status --json` backend rows no longer include the `lifecycle` field or the umbrella columns that accompanied it.

! A `config.yaml` block with a `backend.lemonade:` key is now rejected with an error; the `--lemonade` CLI flag and `LLAMASTASH_LEMONADE` environment variable are removed — migrate to a `backend.generic` entry per RECIPES.md § Lemonade.

! The `PP/S` column is inserted between `TOK/S` and `TOTAL` in the `llamastash requests` table; fields after `TOK/S` have shifted, so `awk -F\t` scripts that addressed those columns by position will read the wrong fields.

! The request summary is split into two rows (counts first, then speeds and token totals), and the field formerly named `Prefill tok/s` is now `Avg PP tok/s`.

WAS THIS USEFUL? 👍 👎

camelid

Feedback ⓘ v0.8.0 NOTES ↗ RANK

Camelid: a Rust-native local inference backend with evidence-gated model compatibility.

> camelid v0.8.0 introduces a persistent Knowledge Library with hybrid retrieval and expanded document support, alongside opt-in cross-chat Memory, GPU min-p sampling, and CUDA inference engine improvements.

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

01 **Knowledge Library with hybrid retrieval and document support**

NEW 75

camelid adds a persistent Knowledge Library where documents are organized into collections that can watch local folders for changes. Hybrid retrieval combines keyword and semantic search (when an encoder is available) across attached documents, selected collections, or the whole library, with verified citations that open the source text. The library supports additional document formats including HTML and source code, with improved PDF fallback, section-aware chunking, and semantic indexing via CUDA and Apple Silicon Metal paths.

— Mechanism and scope described but no exact command or config shown v0.8.0

THINNER COVERAGE BELOW

02 **Opt-in cross-chat Memory for facts and preferences** NEW 50

camelid adds opt-in Memory to persist facts and preferences across chats — off by default, every suggestion requires approval, and saved facts can be edited or forgotten.

— Behavior described clearly but no UI path or config key given v0.8.0

03 **GPU min-p sampling with CPU fallback** NEW 50

camelid adds opt-in GPU min-p sampling with guarded CPU fallback, seeded replay, and KV-history regression checks.

— Names mechanism but no flag or setting to enable it v0.8.0

04 **CUDA conversation cache reuse and cancellation handling**

IMPROVED 30

camelid supports reuse of CUDA conversation caches and responds to cancellation during prompt processing as well as generation.

— Brief description with no metrics or exposed controls v0.8.0

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

oxideterm

Feedback ⓘ v2.2.6 NOTES ↗ RANK

AI-native workspace for local shells and remote machines.Zero Webview, zero OpenSSL, zero telemetry, and no app subscription.Written in Rust+GPUI

> OxideTerm's biggest window added standalone SSH tunnels, a terminal-to-IDE workflow, command-output folding and unified search, alongside privilege-escalation credential matching with 2FA, Telnet/XMODEM tuning, and a large background-customization system (video packs, generated effects, parallax, camera motion, all-day lighting, alignment) with explicit cloud sync direction controls; a Windows 2.2.5 regression wrote CRLF line endings into remote shell-integration scripts, now fixable via a Repair action.

└──▷ **WHAT SHIPPED · 21 FEATURES** most completely described first ? what's the number?

01 **Cloud sync direction controls** NEW 85

Adds a read-only View Cloud Configuration entry to inspect remote configuration before applying it, plus explicit cloud sync direction controls — Use Cloud Configuration , Merge Local and Cloud , and Update Cloud from This Device — with matching explanations and confirmation buttons.

Check what configuration is stored in the cloud before overwriting local settings on a new machine.

📍 In OxideTerm, open Settings › Cloud Sync and select View Cloud Configuration to inspect the remote config read-only, then choose Use Cloud Configuration to download and apply it.

— Every named control given with exact UI steps. v2.2.3

02 **Convert terminal to remote IDE session** NEW 80

Adds Convert to IDE from an existing terminal, inheriting the current user, container, and directory, with a return path to the original terminal; remote IDE access supports SFTP or SSH (pure Shell or on-demand agent).

Open a remote IDE session from an active terminal, inheriting the current user and working directory — useful for editing files on a restricted server without re-authenticating.

📍 In an active terminal session, open the context menu or activity rail and choose Convert to IDE . OxideTerm reuses the current shell environment (user, container, directory) and opens the remote IDE. Choose SFTP or SSH file access in the themed dialog that appears. To return, select Return to Terminal from the IDE toolbar.

Convert an active SSH terminal session directly into a remote IDE session, inheriting the current user, container, and working directory — skip re-navigating to the right path in a separate IDE window.

> 📍 In an active terminal, open the context menu or command bar and select `Convert to IDE`. Choose SFTP or SSH (pure Shell or agent) for file access. The IDE opens in the same environment; use `Return to Terminal` to go back.

— Named menu action and exact return path given. v2.2.6 v2.2.5

---

03 **Standalone SSH tunnels with sync and import/export** NEW

75

Adds standalone SSH tunnel configuration supporting local, remote, and dynamic SOCKS5 forwarding, with cloud sync and encrypted `.oxide` import/export; tunnels can start with the application and run without a terminal pane.

Set up a persistent SOCKS5 dynamic SSH tunnel that starts with OxideTerm — useful for routing traffic through a bastion without keeping a terminal pane open.

> 📍 In OxideTerm, go to SSH Tunnels › Add Tunnel, set Type to `Dynamic SOCKS5`, configure the remote host and local port, enable `Start with application`, then save. The tunnel runs in the background and can be exported via File › Export as encrypted .oxide.

Set up a standalone SOCKS5 dynamic SSH tunnel that starts with the application, without keeping a terminal pane open — useful for proxying traffic through a jump host.

> 📍 In OxideTerm, go to the SSH Tunnels section, create a new tunnel, set the type to Dynamic SOCKS5, configure the local port and remote SSH host, enable `Start with application`, then save. The tunnel appears in the tunnel list and can be started or stopped independently of any terminal.

— Names forwarding types and .oxide format; UI path given, no CLI. v2.2.6 v2.2.5

---

04 **Command-output folding** NEW

75

Adds opt-in command-output folding that collapses completed command output while retaining a summary, exit code, and expand action; fold ranges survive window resizing. Off by default, enabled via Settings > Terminal > Shell Integration.

Enable command-output folding to keep a tidy terminal during long-running tasks — collapse completed command output to a one-line summary while retaining exit status.

> 📍 In OxideTerm Settings, navigate to Terminal › Shell Integration and enable `Command Output Folding` (off by default). After enabling, completed commands in the terminal show a collapse toggle in the gutter alongside the exit code summary.

Enable command-output folding to keep your terminal uncluttered during long build or deployment runs — collapse completed command output while retaining the exit status.

> 📍 In OxideTerm Settings, navigate to Terminal › Shell Integration and enable command-output folding (off by default). Completed commands will show a collapsible summary with exit status and an expand action in the gutter.

— Exact settings path and default state given. v2.2.6 v2.2.5

---

05 **Windows 2.2.5 CRLF shell-integration script bug** BREAKING

75

Windows builds of 2.2.5 may write CRLF line endings into Shell integration scripts, causing remote Linux Bash to fail when loading them; after upgrading to 2.2.6, run Shell integration Update or Repair on affected hosts — updating the application alone does not repair scripts already written to remote hosts.

— Exact remediation action and cause named. v2.2.5

---

06 **Unified search across terminals, IDE, and SFTP editors** IMPROVED

70

Unifies search controls across terminals, the IDE, and SFTP editors with case sensitivity, regular expressions, and multiline input; editors gain capture-group replacement and Replace All.

— Mechanism and scope named but no exact UI path. v2.2.6

---

07 **On-demand background packs via marketplace** NEW 70

Adds a Background Media marketplace category and a Get More Backgrounds entry in the media library so video assets can be installed on demand, independently of the main application.

Install the optional OxideTerm Backgrounds 1.0.0 video pack to get the Dawn Mist and Night Flow 4K background loops without bundling them in the main app.

> 📍 In OxideTerm, open the Plugin Marketplace, select the Background Media category, locate OxideTerm Backgrounds 1.0.0 , and click Install. The pack's media appears in the media library under Get More Backgrounds .

— Named marketplace category and exact UI entry. v2.2.3

---

08 **Seven new generated background effects** NEW 70

Adds seven new generated background effects: subtle texture, fog glow, light tides, combined tides and particles, meteors, depth particles, and water caustics — each with per-effect strength, color, size, speed, direction, roughness, sheen, or density controls.

— All seven effects and controls enumerated, no exact path. v2.2.3

09 **Background parallax and slow camera motion** NEW 70

Adds background parallax so static images and effect layers shift slightly with mouse movement, with depth-particle layers moving by different amounts while terminal text stays fixed, plus slow camera motion for static images with pan, zoom-in, and zoom-out modes, adjustable amount and speed, and edge coverage throughout the repeating motion.

— Mechanism for both effects described, no exact settings path. v2.2.3

10 **Quick-command bar group selector and icon colors** IMPROVED 65

Improves the quick-command bar with a fixed group selector that includes search and persists its selection across restarts independently of management-panel filtering, and adds custom colors to quick-command group icons with validation, reset to theme color, propagation to member commands, and persistence through save, import/export, and cloud sync.

— Two concrete UI additions described but no exact path. v2.2.6

11 **SSH privilege-escalation authentication** NEW 65

Adds SSH privilege-escalation authentication: credential matching with wildcard host support that selects in order of connection-specific, host-matched, and general-default (with unique matches auto-filling and submitting), plus two-factor privilege authentication with independent prompt matching for password-only, password-then-TOTP, and combined password+TOTP response modes.

— Matching order explained, no exact config field named. Modes named but no setup steps given.

v2.2.6

12 **Native video background decoding** NEW 65

Adds native video background decoding via macOS system media, Windows Media Foundation, and Linux GStreamer, routing buffers or textures directly into the graphics backend to reduce CPU copies.

— Platform backends named, no user-facing steps. v2.2.3

13 **Pause Effects While Typing setting** NEW 65

Adds an optional Pause Effects While Typing setting that suspends animated media and effects during continuous input, resuming approximately one second after typing stops.

— Named setting and exact resume timing given. v2.2.3

14 **All-day lighting for backgrounds** NEW 65

Adds all-day lighting that gradually changes background light and color with local time, with morning, afternoon, evening, and night previews.

Enable all-day lighting so your terminal background gradually shifts with the time of day without manually toggling themes.

> 📍 In OxideTerm, open Settings › Appearance › Background, enable `All-Day Lighting`, and use the morning/afternoon/evening/night preview buttons to verify the color transitions before confirming.

— Preview states named; exact UI path shown in example. v2.2.3

---

15 **Nine-point background alignment and display options** NEW

65

Adds nine-point alignment for backgrounds to control which region remains visible after cropping; media opacity, blur, readability overlay, scope, and enabled pages are each independently configurable.

— Controls named, no exact navigation path given. v2.2.3

---

16 **Follow System Appearance light/dark profiles** NEW 60

Adds `Follow System Appearance` support for fully separate light and dark profiles, each carrying an independent application theme, terminal theme, and background — switching without reconnecting sessions.

— Behavior described, no exact settings path shown. v2.2.3

THINNER COVERAGE BELOW

17 **Terminal triggers on matched text** NEW 55

Adds terminal trigger support for click actions, menu entries, and hover hints on matched text, with matched content and named capture groups usable in text and addresses.

— Mechanism named, no configuration steps shown. v2.2.6

---

18 **Title-bar tab scope selector** NEW 50

Adds a title-bar scope selector to filter tabs by all tabs, global pages, local terminals, or a specific connection, with visible/total counts and a `Show All` action.

— Describes UI behavior but no exact navigation path given. v2.2.6

---

19 **Telnet newline config and XMODEM block size** NEW 50

Adds Telnet send/receive newline configuration and XMODEM upload block-size selection (128 or 1024 bytes).

— Concrete values given but no UI path or steps. v2.2.6

20 **RDP plugin version requirement for display fixes** BREAKING

50

RDP display and size renegotiation fixes require RDP plugin 0.2.0, which in turn requires OxideTerm 2.2.5 or later.

— Version numbers given but no install steps. v2.2.5

21 **OS certificate trust store for network requests** NEW

40

Adds network requests using the OS certificate trust store, enabling enterprise and proxy CA certificates installed on the host.

— Benefit stated but no mechanism or steps detailed. v2.2.5

NEW INTEGRATION Xiaomi MiMo provider

NEW INTEGRATION DeepSeek web search and page-reading tools

UPDATED INTEGRATION Image input support for AI provider interfaces

└──▷ BREAKING ON UPGRADE

! Windows builds of 2.2.5 may write CRLF line endings into Shell integration scripts, causing remote Linux Bash to fail when loading them; after upgrading to 2.2.6, run Shell integration Update or Repair on affected hosts — updating the application alone does not repair scripts already written to remote hosts.

! RDP display and size renegotiation fixes require RDP plugin 0.2.0, which in turn requires OxideTerm 2.2.5 or later.

v2.2.4 NOTES ↗ STABLE

OxideTerm 2.2.4 adds AI image input, a common SSH privilege-credential scope, per-device command history, offline port-forwarding management, and SFTP Go to Terminal Directory.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.2.4
https://github.com/AnalyseDeCircuit/oxideterm.git
# already have the repo? check out this version:
$ git checkout v2.2.4
```

Adds AI image input to conversations via clipboard paste or file selection, with removable thumbnails, per-conversation storage, and support across OpenAI Chat Completions, Responses, Anthropic, and Gemini interfaces.

Adds a common SSH privilege-credential scope reusable for sudo, su, and custom password prompts, with connection-specific matches taking priority.

Adds optional per-device command history for saved SSH, Telnet, Mosh, and serial connections, persisting across restarts, with up to 10,000 entries across saved connections and configurable retention (default 90 days).

Saved port-forwarding rules remain visible, editable, saveable, and deletable while disconnected; `Save and Start` uses the existing SSH connection flow without opening an extra terminal.

Adds `⌘⇧G` (macOS) / `Ctrl+Shift+G` (Windows/Linux) shortcut for `Go to Terminal Directory` in SFTP, navigating to the current or most recently used terminal directory on the same server, including detached SFTP windows.

+3 more

WAS THIS USEFUL? 👍 👎

liliang-cn CortexDB

Feedback ⓘ v2.139.0 NOTES ↗ RANK

AI memory and a knowledge graph in one SQLite file. Pure Go: vectors, RAG, agent memory, RDF/SPARQL, Cypher, 80+ MCP tools.

> CortexDB's latest releases add a source fact-checking tool, batch ID lookup, and document lifecycle controls including content-hash deduplication, expiry sweeping, and review-status gating.

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

01 **Content-hash deduplication for document ingestion** NEW

65

Adds content-hashing to `ingest_document` so identical re-ingests are skipped and duplicates can be refused.

— Names the exact tool and describes the dedup mechanism. v2.137.0

02 **Batch ID lookup for knowledge_get** IMPROVED

60

Extends `knowledge_get` to accept a list of IDs, matching the existing behavior of `memory_get`.

— Names both tools and the exact change made. v2.139.0

THINNER COVERAGE BELOW

03 **Source fact-checking tool** NEW

50

Adds `check_against_sources`, a new tool that identifies facts in a sentence that its cited sources do not contain.

— Names the tool but gives no mechanism or example. v2.139.0

04 **Confined key graph walks via authorization layer** NEW

40

Enables confined keys to walk the graph, constrained by the facade, via the authorization layer.

— Explains scope but no named API or config surface. v2.139.0

05 **Document expiry with automated sweeping** NEW

40

Adds document expiry with an automated sweep that removes expired documents.

— Describes mechanism but no field or schedule details. v2.137.0

06 **Review-status gating for documents and memories** NEW

40

Adds a review status field to documents and memories, with the ability to hide items based on that status.

— Explains the feature but no field name or API given. v2.137.0

07 **Automatic key file reload** IMPROVED 35

Servers now re-read the key file automatically when it changes, without requiring a restart.

— Clear behavior but no path or config key named. v2.139.0

08 **Richer request logging of tool arguments** IMPROVED 25

The request log now records the row names a tool call names in its arguments, improving auditability of tool invocations.

— Describes the change but no field names or log format given. v2.139.0

WAS THIS USEFUL? 👍 👎

knowhere

Feedback ⓘ v1.2.30 NOTES ↗ RANK

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

> Knowhere shipped a retrieval query mode router, a unified chunk metadata schema for downstream retrieval and graph construction, and parsing improvements including preserved document-root opening pages and new taxonomy metadata.

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

01 **Unified chunk metadata schema** IMPROVED 37

Chunk relations, importance scores, and entities-first metadata are unified into a single schema, enabling consistent downstream retrieval, hierarchy construction, and graph construction.

– Describes scope and purpose but no schema fields or API shown v1.2.26

02 **Parsing: preserved opening pages and new metadata** IMPROVED 35

Parsing now preserves opening pages at the document root, and adds IPTC domain taxonomy, TOC keywords, and image region metadata.

– Names several metadata types but no fields or usage shown v1.2.27

03 **Retrieval query mode router** NEW 33

A new router dispatches retrieval queries to either an agent or classic retrieval path, selecting the mode automatically.

– Names two modes but no config or API to invoke them v1.2.30

WAS THIS USEFUL? 👍 👎

fabio-rovai open-ontologies

Feedback ⓘ v2.1.0

NOTES ↗ RANK

open-ontologies is a tool to plan, apply and roll back changes to a production ontology, with a blast radius report and a proof an auditor can re-check.

> open-ontologies v2.1.0 adds named tool profiles for scoping the CLI surface, a crosswalk-probe for testing subsumption clashes before merge, reasoning-trace labelling, OWL 2 functional syntax support, and changes the default align scoring algorithm (breaking).

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

01 **Named tool profiles for CLI/client exposure** NEW 90

Adds the `--tool-profile` flag, `OPEN_ONTOLOGIES_TOOL_PROFILE` environment variable, and `profile` key under `[tools]` config to expose a named subset of the tool surface to a client. Nine profiles are available, and the system reports `compiled_in`, `servable_in_this_build`, and `exposed` counters.

— Exact flag, env var and config key with concrete counters named v2.1.0

02 **Reasoning trace labelling via onto_trace_label** NEW 90

Adds `onto_trace_label` and the `trace-label` verb to label each step of a reasoning trace as `entailed`, `not_entailed`, or `outside_the_fragment`, accepting traces in `oo-trace/1` format read from `derivations.tsv`.

— Named verb, label values and exact input format and file v2.1.0

03 **crosswalk-probe for clash detection** NEW 85

Adds the `crosswalk-probe` CLI subcommand and batch verb to carry a crosswalk's subsumptions into the target and report added clashes, exiting 1 on a contradiction.

— Named subcommand, verb and exact exit-code behaviour v2.1.0

04 **OWL 2 functional syntax (OFN) loading** NEW 85

Adds OWL 2 functional syntax loading across `load`, `validate`, `convert`, `lint`, `batch`, `onto_load`, `onto_validate`, and `onto_convert`; OFN is detected by a `Prefix(` or `Ontology(` first token and `.ofn` files are accepted by extension.

— Names every command, detection tokens and file extension v2.1.0

05 **Align scoring algorithm changed (breaking)** BREAKING 75

The default `align` scoring algorithm has changed, raising the auto-apply threshold from 0.85 to 0.95 on the new scale. Use the new `--scoring legacy` flag on `align` to reproduce the prior algorithm's results.

— Names flag and exact old/new threshold values v2.1.0

THINNER COVERAGE BELOW

06 **Human-readable SELECT query output** NEW 55

Adds a `--human` flag that renders native SELECT query results as tables, including empty results.

— Named flag with clear, narrow behaviour v2.1.0

07 **Many-to-one cardinality option for align** NEW 45

Adds `--cardinality many-to-one` to `align` to keep several source terms mapped onto one coarser target term.

— Named flag but no further mechanism given v2.1.0

└──▷ **BREAKING ON UPGRADE**

! The default `align` scoring algorithm has changed; the auto-apply threshold moves from 0.85 to 0.95 on the new scale. Use `--scoring legacy` to reproduce previous results.

WAS THIS USEFUL? 👍 👎

python-sdk

Feedback ⓘ v10.2.0 NOTES ↗ RANK

Official Python SDK for the Pinecone vector database

> Pinecone's Python SDK adds full-text filtering and server-side query/document embedding, alongside a long list of stricter validation errors and other breaking changes to error handling and the documents API.

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

01 Server-side text embedding for documents and queries NEW

95

SchemaBuilder.add_string_field() gains `embed=True` and `sparse_embed=True` flags so Pinecone embeds plain-text document fields server-side, with the model configured via `FieldEmbedConfig` or `FieldSparseEmbedConfig`, storage suppressed with `store_text=False`, and token consumption tracked in `usage.embed_total_tokens`. Search queries can now be embedded server-side too via new `EmbedQuery` and `SparseEmbedQuery` clause types in `score_by`, each of which must be the sole clause in that list.

Create an index whose `body` field Pinecone embeds for you, then upsert plain-text documents and search with a natural-language query — no vectors to manage.

```python
from pinecone import EmbedQuery, Pinecone, SchemaBuilder

pc = Pinecone()  # reads PINECONE_API_KEY

schema = SchemaBuilder().add_string_field("body",
embed=True).build()
pc.indexes.create(name="articles", schema=schema)
index = pc.index("articles")

index.documents.upsert(
    namespace="en",
    documents=[
        {"_id": "a1", "body": "Aqueducts carried water by
gravity"},
        {"_id": "a2", "body": "The Colosseum seated 50,000
people"},
    ],
)

results = index.documents.search(
    namespace="en",
    top_k=2,
    score_by=[EmbedQuery(field="body", query="how did Rome move
water")],
)
for doc in results.matches:
    print(doc.id, doc.score)
```

— Full mechanism, config options and runnable example given `v10.1.0`

02 **Stricter input validation and error types across the SDK**

90

BREAKING

Unknown keys in a dict `score_by` clause now raise `PineconeValueError` instead of being silently ignored; some former `ValueError`s are now `PineconeTypeError`, so code catching `ValueError` must catch `PineconeError` or `TypeError` instead. Documents methods now require a dict `filter=`, so filter-builder `Condition` objects must be converted with .to_dict() first. A zero, negative, or NaN `timeout` or `total_timeout` now raises `PineconeValueError` before the request is sent; a malformed `score_by`, `top_k`, or `limit` likewise raises `PineconeTypeError` first. A bare string where a list is expected now raises `PineconeTypeError` instead of being silently split into characters. A negative wait `timeout` on `indexes.create` or `indexes.delete` now raises (`-1` still skips waiting), a non-string name or ID in a URL now raises `PineconeTypeError` (callers must pass str(...)), an API key with whitespace

or non-ASCII characters now raises on client construction, and schema field names over 64 UTF-8 bytes now raise in pc.indexes.create(schema={...}).

— Every changed type, method and condition named verbatim v10.1.0

03 **Full-text filter methods on Field** NEW 75

`Field` gains `match_phrase`, `match_all`, and `match_any` methods for filtering `documents.search` and `documents.fetch` results by words in a text field: `match_phrase` requires the words in order, `match_all` requires every word in any order, and `match_any` requires at least one word.

Narrow a semantic document search to only results whose text field contains a specific phrase and whose numeric field meets a threshold.

python

```python
from pinecone import EmbedQuery, Field, Pinecone

pc = Pinecone()
index = pc.index("articles")  # 'body' is a string field
declared with embed=True

results = index.documents.search(
    namespace="en",
    top_k=5,
    score_by=[EmbedQuery(field="body", query="how did Rome move
water")],
    filter=(Field("body").match_phrase("roman aqueduct") &
Field("year").gte(100)).to_dict(),
)
```

— Clear semantics per method plus runnable filter example v10.2.0

04 **Write visibility now reported via documents.search** BREAKING 65

To wait for a write to become visible, callers must now poll `documents.search`, which reports `is_reconciled`; `documents.fetch` no longer reports `is_reconciled`.

— Clear before/after and migration path given v10.1.0

05 **PINECONE_ADDITIONAL_HEADERS no longer logged** BREAKING 60

`PINECONE_ADDITIONAL_HEADERS` is no longer logged; users should check old logs and rotate any exposed credentials.

— Names the env var and gives a concrete remediation step v10.1.0

THINNER COVERAGE BELOW

06 **PayloadTooLargeError exception type** NEW 50

The SDK now raises `PayloadTooLargeError`, a subclass of `ApiError`, on HTTP 413 responses, giving callers a distinct exception to catch instead of a generic API error.

— Names the type and trigger but no usage example v10.1.0

07 **Higher fetch limit for filtered document fetches** IMPROVED 45

`documents.fetch` now accepts a `limit` of up to 10,000 for filtered fetches.

— Concrete limit named but no example v10.1.0

08 **Automatic batch splitting in batch_upsert** IMPROVED 40

`documents.batch_upsert` now automatically splits a batch the server rejects as too large instead of failing the whole request.

— Behaviour explained but is automatic, nothing to configure v10.1.0

09 **Python 3.15 support** NEW 30

The SDK adds Python 3.15 support, with prebuilt wheels.

— Bare compatibility note with no further detail v10.1.0

└──▷ BREAKING ON UPGRADE

! To wait for a write to be visible, poll `documents.search` (which reports `is_reconciled`), not `documents.fetch`; `documents.fetch` no longer reports `is_reconciled`.

! Unknown keys in a dict `score_by` clause now raise `PineconeValueError` instead of being silently ignored.

! Some `ValueError`s are now `PineconeTypeError`; code catching `ValueError` must be updated to catch `PineconeError` or `TypeError`.

! Documents methods now require a dict `filter=`; filter-builder `Condition` objects must be converted with .to_dict() first.

! A zero, negative, or NaN `timeout` or `total_timeout` now raises `PineconeValueError` before the request is sent.

! A bare string where a list is expected now raises `PineconeTypeError` instead of being silently split into characters.

! A malformed `score_by`, `top_k`, or `limit` now raises `PineconeTypeError` before the request is sent.

! A negative wait `timeout` on `indexes.create` or `indexes.delete` now raises; `-1` still skips waiting.

! A non-string name or ID in a URL now raises `PineconeTypeError`; callers must pass str(...).

! An API key with whitespace or non-ASCII characters now raises when the client is constructed.

! Schema field names over 64 UTF-8 bytes now raise in pc.indexes.create(schema={...}).

! `PINECONE_ADDITIONAL_HEADERS` is no longer logged; check old logs and rotate any exposed credentials.

WAS THIS USEFUL? 👍 👎

Volcengine OpenViking

Feedback ⓘ v0.5.2 NOTES ↗ RANK

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

> OpenViking v0.5.2 adds a new PATCH endpoint for managing Watches, privacy-tree MCP tooling, durable content-write workflows, and expanded Compile capabilities, while also shipping several breaking changes — a LangChain integration package split, Working Memory off by default, MCP tool renaming, and a peer-identity ID format change.

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

01 Watch update via PATCH endpoint

BREAKING

90

Adds `PATCH /api/v1/watches/{task_id}` to update or reactivate an existing Watch's schedule without deleting and recreating it. This replaces the prior behavior where re-importing with `watch_interval > 0` would update or reactivate a Watch — that path no longer does so, so callers must use the PATCH endpoint or delete the old Watch before creating a replacement.

Update an existing Watch's schedule without deleting and recreating it, using the new PATCH endpoint.

```
$ curl -X PATCH https://<host>/api/v1/watches/<task_id> \
  -H 'Authorization: Bearer <user-or-admin-key>' \
  -H 'Content-Type: application/json' \
  -d '{"watch_interval": 3600}'
```

— Exact endpoint, verb, and a runnable curl example provided. v0.5.2

02 MCP tool names renamed to openviking_* prefix

BREAKING

85

Pi extension 0.4.0 renames all MCP tool names from `viking_*` to `openviking_*`; `--tools` and `--exclude-tools` entries must be updated accordingly, and root API keys are now refused on `/mcp` — use a user or admin key instead.

— Exact flags, prefixes, and key requirements given. v0.5.2

03 LangChain integration moved to separate package

BREAKING

80

The server no longer ships `openviking.integrations.langchain` or its forwarding modules; install `langchain-openviking` (or `langchain-openviking[langgraph]` for

LangGraph support) and replace all import prefixes with `langchain_openviking`.

— Exact package names and import migration steps given. v0.5.2

04 **Working Memory disabled by default** BREAKING 75

Working Memory (WM/checkpoint summaries) is now off by default; callers that require it must explicitly pass `enable_working_memory=true` and update relevant agent plugins.

— Exact flag and required caller action given. v0.5.2

05 **Lossless external peer identity IDs** BREAKING 65

Mixed-script external peer identities now use lossless `ext-<base64>` IDs; older lossy peer directories are not read automatically and require operator intervention to migrate.

— Names the ID format but migration steps are vague. v0.5.2

THINNER COVERAGE BELOW

06 **Session privacy extraction and privacy-tree MCP tools** NEW 50

Adds automatic session privacy extraction and restore flows, new privacy-tree MCP tools, and the ability to inspect the current privacy configuration tree.

— Describes capability but no flag, endpoint, or config key given. v0.5.2

07 **OAuth refs in LangChain document imports** NEW 40

Document imports now support OAuth refs in LangChain connectors.

— Names the feature but not a specific field or flag. v0.5.2

08 **Incremental updates, scheduling, and history in Compile** IMPROVED 35

Compile workflows now support incremental updates, scheduling, and version history.

— Three additions named but no config or API surface given. v0.5.2

09 **CLI defaults to Seed 2.1 Lite VLMs** IMPROVED 35

CLI initialization now defaults to Seed 2.1 Lite VLMs.

— Names the new default version with no further detail. v0.5.2

10 **Durable content-write task workflow** NEW 30

Adds a durable content-write task workflow for improved server-side reliability.

— Reliability claim given with no mechanism or surface named. v0.5.2

11 **Gemini Embedding 2 support on server** NEW

30

Adds server-side support for Gemini Embedding 2.

— Names the model but no configuration detail given. v0.5.2

╰──▷ BREAKING ON UPGRADE

! The server no longer ships `openviking.integrations.langchain` or its forwarding modules; install `langchain-openviking` (or `langchain-openviking[langgraph]` for LangGraph) and replace all import prefixes with `langchain_openviking`.

! Working Memory (WM/checkpoint summaries) is off by default; callers that require it must now explicitly pass `enable_working_memory=true` and update relevant agent plugins.

! Re-importing with `watch_interval > 0` no longer updates or reactivates an existing Watch; use `PATCH /api/v1/watches/{task_id}` or delete the old Watch before creating a replacement.

! Pi extension 0.4.0 renames all MCP tool names from `viking_*` to `openviking_*`; `--tools` and `--exclude-tools` entries must be updated, and root API keys are refused on `/mcp` — use a user or admin key instead.

! Mixed-script external peer identities now use lossless `ext-<base64>` IDs; older lossy peer directories are not read automatically and require operator intervention to migrate.

WAS THIS USEFUL? 👍 👎

fastembed-rs

Feedback ⓘ v7.5.0 NOTES ↗ RANK

Rust library for generating vector embeddings and reranking locally!

> fastembed-rs added local Qwen3-Reranker support with throughput tuning, split the `qwen3` Cargo feature into `qwen3-text` / `qwen3-vl` while making ONNX Runtime optional, expanded Qwen3 embedding options with layer pooling, output-size control and padding-free batching, added GGUF file loading for quantized Qwen3 embeddings, and shipped extensive Metal GPU performance optimizations for the Candle backend.

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

01 Qwen3 Cargo feature split into qwen3-text and qwen3-vl

BREAKING

90

Splits the `qwen3` feature into `qwen3-text` and `qwen3-vl` Cargo features, separating text-only Qwen3 embedding models from the multimodal Qwen3VLEmbedding vision-language model. The `qwen3` Cargo feature is replaced by these two; any `Cargo.toml` entry specifying `features = ["qwen3"]` must be updated.

— Exact feature names and migration instruction make this fully actionable. v7.4.0

02 Metal GPU performance optimizations for Qwen3 on Candle backend

IMPROVED

81

Accelerates Qwen3 Metal inference with a fused q/k norm-and-RoPE kernel, reordered attention output, and optimized last-layer pooling. Routes Qwen3 attention through a fused SDPA kernel on Metal for both padded and unpadded batches, passing q, k, and v tensors as strided views to eliminate intermediate copies. Runs Qwen3 RMSNorm and RoPE through Candle's fused kernels, applies a fused softmax in attention (nearly halving peak memory usage on long texts), and accelerates the MLP activation (`silu(gate) * up`) as a single fused Metal kernel.

— Enumerates every named kernel optimization with concrete mechanism and effect. v7.2.0 v7.4.0

03 Qwen3-Reranker support and throughput optimization

NEW

67

Adds Qwen3-Reranker support in the Candle backend for local reranking with Qwen3 models. The shared prompt head now runs once per `score` call instead of redundantly per candidate, improving reranking throughput.

— Names model, backend and the optimized call, but no exact API shown. v7.3.0 v7.5.0

04 **GGUF file loading for Qwen3 embedding models** NEW 65

Adds support for loading Qwen3 embedding models from GGUF files in the Candle backend, enabling quantized local inference without ONNX conversion.

— Clear capability and benefit, no exact loading API shown. v7.2.0

05 **Qwen3 embedding options: layer pooling, output size, padding-free batching**

NEW 61

Adds pooled Qwen3 embedding vectors from chosen layers in a single forward pass via the Candle backend, an output size option for shorter Qwen3 embedding vectors (dimensionality reduction at inference time), and equal-length grouping for Qwen3 text batches that eliminates padding overhead across the batch.

— Three mechanisms named but no exact flags or config keys given. v7.3.0

THINNER COVERAGE BELOW

06 **ONNX Runtime made optional dependency** NEW 48

Makes ONNX Runtime an optional dependency, allowing the library to be used without it when using candle-based backends such as `qwen3`.

— Explains the change but no exact Cargo feature flags to toggle it. v7.4.0

└──▷ **BREAKING ON UPGRADE**

! The `qwen3` Cargo feature is replaced by `qwen3-text` (for text embedding models) and `qwen3-vl` (for `Qwen3VLEmbedding`); any `Cargo.toml` entry specifying `features = ["qwen3"]` must be updated.

WAS THIS USEFUL? 👍 👎

InfiniFlow RAGFlow

Feedback ⓘ v0.27.3 NOTES ↗

RAGFlow is a retrieval-augmented generation engine built on deep document understanding, combining chunk-level citations with agentic workflows.

Adds `document_ids` retrieval filtering, `image_update_mode` API field, claim/evidence pipeline fields, HTML fragment parsing, and ports agentic RAG to Go.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.27.3
https://github.com/infiniflow/ragflow.git
# already have the repo? check out this version:
$ git checkout v0.27.3
```

Adds `document_ids` parameter to the retrieval node and retrieval tool in the agent, enabling filtering of retrieval results to specific documents.

Adds `image_update_mode` field to the chunk image update API.

Adds claim/evidence fields to the tree/pageindex compilation pipeline.

Rerankings now operate on natural chunk text instead of tokenized `content_ltks`, improving relevance quality.

UPDATED INTEGRATION Supports HTML fragment parsing in addition to full HTML documents, and allows customizing the file extension when saving database fields.

+2 more

NEW INTEGRATION Feishu Wiki file connector (Python backend)

NEW INTEGRATION Feishu Wiki connector for Go backend

NEW INTEGRATION AnonRouter model provider

NEW INTEGRATION API-Route model provider

UPDATED INTEGRATION WeCom channel can now bind an agent

UPDATED SUPPORTED PLATFORM Cache backend migrated from Valkey 8 to apache/kvrocks:2.16.0

└──▷ BREAKING ON UPGRADE

! Cache backend is replaced: Valkey 8 is removed in favor of `apache/kvrocks:2.16.0`, requiring a cache backend migration on upgrade.

WAS THIS USEFUL? 👍 👎

AWS Context Ontology Accelerator

Feedback

v0.3.7 CODE ↗ RANK

Context Ontology Accelerator provides an open-source semantic layer that helps AI agents make consistent, explainable decisions using formal ontologies and knowledge graphs.

> Context Ontology Accelerator v0.3.7 adds idempotent retry support for API Gateway create operations, new query tuning options, and richer label capture during ontology induction.

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

01 Idempotency tokens for create-operation retries NEW

90

Create operations now accept a `clientToken` field in the JSON request body, enabling safe retries through an idempotency store that replays the original 2xx response on re-submission, returns `409` on conflicting re-use of a token, and returns `400` for malformed tokens.

Safely retry a create operation from a client without risking duplicates — include `clientToken` so the service replays the original response on re-submission.

```
$ curl -X POST https://<api-gateway-host>/<stage>/<resource> \
  -H 'Content-Type: application/json' \
  -d '{"clientToken": "550e8400-e29b-41d4-a716-446655440000",
...}'
```

— Names exact field, response codes, and includes runnable curl example v0.3.7

02 New Query request options: tierOverride, timeoutMs, maxResults

NEW

65

The `Query` request body gains optional `tierOverride`, `timeoutMs`, and `maxResults` members, which are validated against configured min/max contract bounds before being forwarded to the ontology engine.

— Names all three fields and validation rule, but no usage example v0.3.7

03 Alternate label capture during ontology induction IMPROVED

60

During unstructured ontology induction, the service now emits `skos:altLabel` triples for every observed alternate label form of a class IRI, alongside the existing `rdfs:label` for the first-seen preferred form.

— Explains mechanism and vocabulary used but no example or config shown v0.3.7

WAS THIS USEFUL? 👍 👎

rabitq-library

Feedback ⓘ v0.7.1 NOTES ↗ RANK

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

> RaBitQ Library's two releases restructured its C++ core into modular, independently linkable components with a stable v1 API, and added more granular per-cluster storage for IVF indexes.

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

01 **Modular C++ core split with stable v1 API** NEW 79

The library is split into separate `rabitqlib::core`, `rabitqlib::fastscan`, and index components. `rabitqlib::core` provides rotation, per-vector encoding, query preparation, and distance estimation as a standalone linkable unit; `rabitqlib::fastscan` builds on top of it to provide batch encoding and lookup-table scoring over 32-vector layouts. `rabitqlib::core` now exposes a stable v1 API, giving downstream engines a long-term-supported quantization surface to build against.

— Names components and API version but no usage example v0.7.0

THINNER COVERAGE BELOW

02 **Per-cluster storage for IVF indexes** NEW 35

Adds per-cluster storage for IVF indexes, enabling more granular on-disk layout for large partitioned indexes.

— Describes benefit but no config key or API detail v0.7.1

WAS THIS USEFUL? 👍 👎

Docling

Feedback ⓘ v2.142.0 NOTES ↗ RANK

Docling parses PDFs, Office files, and HTML into structured documents with layout, tables, and reading order preserved for downstream RAG pipelines.

> Docling v2.142.0 is a performance-and-accuracy patch focused on its AFP backend and internal OCR/image-processing pipeline, with no new user-facing surfaces.

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

01 **AFP line boundary detection accuracy** IMPROVED 42

The AFP backend now treats a move to a new baseline as a line boundary, producing more accurate line segmentation in AFP documents.

— Explains mechanism but no user-facing config or flag. v2.142.0

02 **macOS OCR temporary PNG elimination** IMPROVED 40

Eliminates temporary PNG files during macOS OCR, reducing disk I/O overhead in the OCR pipeline.

— Clear mechanism and platform scope, but internal only. v2.142.0

03 **OCR image crop conversion via numpy.asarray** IMPROVED 40

Converts PIL image crops with `numpy.asarray` in the OCR path for lower-overhead image preparation.

— Names concrete method but no actionable surface for readers. v2.142.0

04 **Image classification scalar read reduction** IMPROVED 15

Reduces scalar reads in image classification for faster throughput.

— Bare performance claim with no mechanism or numbers. v2.142.0

WAS THIS USEFUL? 👍 👎

Qdrant

Feedback ⓘ v1.20.0 NOTES ↗ RANK

Qdrant - High-performance, massive-scale Vector Database and Vector Search Engine for the next generation of AI. Also available in the cloud https://cloud.qdrant.io/

> Qdrant v1.20.0 adds BM25 scoring and a code tokenizer to full-text indexes, nested payload indexes and substring filters, configurable audit log routing, and snapshot storage retry/timeout tuning, alongside a breaking change to internal auth defaults.

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

01 **Configurable audit log output destination** NEW 78

Adds the `audit.output` config key to write audit events to `file` (default), `stdout`, or `stderr`, allowing events to be routed to a log aggregator instead of a file.

Route audit events to stdout for ingestion by a log aggregator instead of writing to a file.

```yaml
service:
  enforce_internal_auth: true

audit:
  output: stdout
```

— Exact config key, values, and YAML example given v1.20.0

02 **Internal auth enforcement now defaults to enabled** BREAKING 75

`service.enforce_internal_auth` now defaults to `true` when an API key is configured — existing distributed deployments with an API key but no internal auth enforcement will now reject unauthenticated internal requests on upgrade.

— Exact config field and upgrade impact described, no remediation steps v1.20.0

03 **BM25 scoring and code tokenizer for full-text indexes** NEW 73

Adds a `code` tokenizer and BM25 scoring directly on full-text payload indexes, improving accurate search over source-code payloads.

Create a full-text index with the new `code` tokenizer to search source code payloads accurately.

```
$ curl -X PUT
'http://localhost:6333/collections/my_collection/index' \
  -H 'Content-Type: application/json' \
  -d '{
    "field_name": "source_code",
    "field_schema": {
      "type": "text",
      "tokenizer": "code"
    }
  }'
```

— Named tokenizer and scoring method with runnable curl example v1.20.0

04 **Substring match filter condition** NEW 73

Adds `match: { substring: "..." }` filter condition for full-scan substring matching on payload fields, usable without a full-text index.

Filter points using a substring match when you need loose, full-scan text matching without a full-text index.

```
$ curl -X POST
'http://localhost:6333/collections/my_collection/points/scroll'
\
  -H 'Content-Type: application/json' \
  -d '{
    "filter": {
      "must": [
        {
          "key": "description",
          "match": { "substring": "exploit" }
        }
      ]
    },
    "limit": 20
  }'
```

— Exact filter syntax and curl example provided v1.20.0

05 **Inline HNSW vector storage** NEW 70

Adds the `hnsw_config.inline_storage` config key to store vectors only inside the HNSW graph file, reducing disk usage.

— Exact config key named with its effect on disk usage v1.20.0

06 **Retry and timeout settings for snapshot storage** NEW

63

Adds `timeout_sec`, `connect_timeout_sec`, `max_retries`, and `retry_timeout_sec` settings to S3, GCS, and Azure snapshot storage configuration.

— All four config keys and providers named, no example v1.20.0

07 **Nested payload indexes for array elements** NEW

60

Adds a `nested` index parameter for per-element payload indexes, enabling faster nested condition filtering on arrays of objects.

— Named parameter but no usage example given v1.20.0

THINNER COVERAGE BELOW

08 **New formula query expressions** NEW

45

Adds `acosh`, `max`, and `min` formula query expressions for custom scoring.

— Named functions but no usage shown v1.20.0

09 **Collection creation timestamp field** NEW

40

Adds a `created_at` field to collection info recording the collection's creation time.

— Named field but no retrieval example v1.20.0

10 **Timeout parameter for get points API** NEW

35

Adds a `timeout` parameter to the get points API.

— Bare mention of parameter, no example or mechanism v1.20.0

11 **Configurable ID tracker memory placement** NEW

33

Adds configurable ID tracker memory placement via memory-tier config.

— Vague mechanism, no config key named v1.20.0

└──▷ **BREAKING ON UPGRADE**

! `service.enforce_internal_auth` now defaults to `true` when an API key is configured — existing distributed deployments with an API key but no internal auth enforcement will now reject unauthenticated internal requests on upgrade.

WAS THIS USEFUL? 👍 👎

stashbase

Feedback ⓘ v2.16.17 NOTES ↗ RANK

An open-source, local-first writing workspace where you can write with Claude Code and Codex, drawing on your own sources and past work.

> StashBase shipped a major round of Agent workflow upgrades — Crew for Default role assignment, Auto/Ask/Edit/Plan approval modes, and a turn-level Markdown review flow that replaces the `suggest_edits` tool with `edit_file` / `write_file` — alongside in-app Plus/Pro subscription management via Stripe, a shared Persona library, broader sign-in options, expanded error diagnostics, and cross-platform native installers.

└──▷ **WHAT SHIPPED · 30 FEATURES** most completely described first ? what's the number?

01 Markdown change review and direct Agent writes

BREAKING

85

Adds a turn-level Markdown review flow accessible via Changed in this turn → Review, with Undo or Keep controls for individual changes or the whole turn; version checks prevent older turns from overwriting newer work. Routes Claude and the default Agent to use `edit_file` for editing and `write_file` for creating project documents, replacing the retired `suggest_edits` proposal/polling/Accept-Reject flow; extends `edit_file` to fill an existing empty file with normal version and project checks applied. Shows Markdown files changed by an Agent turn, letting readers review, keep, or undo individual changes, with Agents writing directly unless a proposal is explicitly requested. Enables reviewing Agent prose revisions inline within documents. The `suggest_edits` tool is removed; external MCP clients must replace calls to it with `edit_file` (to edit) or `write_file` (to create documents).

Review, selectively undo, or keep agent-written Markdown changes before they are committed to a project document.

> 📍 In the StashBase Chat panel, open Changed in this turn, then select Review. Use Undo or Keep on individual changes or click the whole-turn controls to accept or roll back all edits from that turn.

Review, keep, or undo individual Markdown changes made by an Agent turn before they are committed.

> 📍 After an Agent turn completes, open the changed-files panel in stashbase to see each modified Markdown file listed. Select a file, then choose Keep or Undo for each individual change.

— Names exact tools and gives a migration path for the removed tool. v2.15.1 v2.13.2

v2.12.0

02 **Files panel and Quick Open visibility and actions** NEW

80

Adds `Copy path` and `Copy relative path` context actions and `Reveal in Finder / File Explorer` navigation to file and folder context menus in the Files panel, and enables moving files and folders by dragging within the Files tree (open tabs and document links update automatically, and external drops import into the target folder); also shows project configuration and skills directories (`.agents` , `.claude` , `.codex` , `.github` , `.vscode`) in Files and Quick Open by default, independent of the hidden-directory visibility preference.

Quickly copy an absolute or relative path for a file or folder to use in scripts or documentation.

> 📍 In the Files panel, right-click any file or folder, then select `Copy path` for the absolute path or `Copy relative path` for the project-relative path.

Reorganise project assets without breaking open tabs or internal links — drag a folder to its new location in the Files tree and all references follow.

> 📍 In the Files tree, click and drag any file or folder to the target folder. Open tabs and document links update automatically; dropping an external file into a folder imports it there.

Jump to a file's location on disk without leaving the app — useful when you need to hand a path to an external tool.

> 📍 Right-click any file or folder in the Files tree, then select `Reveal in Finder` (macOS) or `Reveal in File Explorer` (Windows).

— Exact context-menu commands named with step-by-step usage. Named directories but no steps beyond default visibility.

v2.16.14 v2.16.17 v2.13.2

03 **Crew for Default chat roles** NEW

80

Adds Crew for Default, letting users assign Writer, Reviewer, Reader, and Humanizer roles — each with its own model choice — to any chat; new chats remember the last selection while existing chats keep their own. Introduces dedicated Crew call cards showing member, model, brief, result, and running/failed/stopped state, with historical calls retaining original attribution.

Assign a dedicated Reviewer and Humanizer with separate model choices to a chat so that review-only requests stay scoped without triggering a full rewrite cycle.

📍 In the chat panel, open Crew for Default and select Reviewer and Humanizer; choose a model for each member. Submit a review-only request — the main assistant delegates review work to the Reviewer; no rewrite cycle starts.

— Detailed roles, states, and attribution with a concrete example. v2.16.0

04 **Sign-in methods and authentication reliability** NEW

75

Adds email-code (six-digit) sign-in as an alternative to Google sign-in, and sign-in method management (connect or disconnect Google) under Settings → Agents → Account and the sidebar account menu, each confirmed by a fresh verification code. Switches desktop Google sign-in, session renewal, and sign-out to Better Auth, with production verification of account and subscription association. Adds failed sign-in error pages that surface the concrete network cause — DNS, TLS, timeout, or proxy connection errors — and authentication now uses the native system proxy configuration, including automatic proxy settings.

Manage which sign-in methods are active on your account without losing your subscription or credits.

📍 Go to Settings → Agents → Account, then choose Connect Google or Disconnect Google. A fresh verification code is sent to your account email to confirm each change.

— Exact settings path, Better Auth switch, and error causes named. v2.16.10 v2.16.15 v2.16.17

05 **Error diagnostics and failure reporting across Agent runtimes**

NEW

75

Correlates Agent failures with their turns and setup attempts, distinguishing successful installation from a remaining authentication requirement. Preserves actionable error messages, stacks, nested causes, provider request IDs, installer output, and runtime context across the app, Agent runtimes, HTTP services, and MCP, with credentials filtered and diagnostic payloads bounded. Captures native startup, process-exit, and updater failures for delivery by the local service, with retry for transient renderer diagnostic delivery failures. Adds early-access error diagnostics for Agent setup and runtime failures, local API and background errors, and renderer/native-operation failures — controlled by the existing Share basic usage statistics preference; automatic reports include controlled summaries, known error codes, and stack line/column locations, while excluding document content, chat content, private paths, and raw logs. Agent connection and update failure details now surface inline at the point of failure, preserving the user's draft and conversation.

Enable or disable both usage statistics and the new error diagnostics reporting with a single toggle.

> 📍 In the stashbase desktop app, go to Settings › Share basic usage statistics and toggle the switch — this single preference controls both usage statistics and automatic error diagnostics reports.

— Extensive mechanism detail and a named preference toggle. v2.15.10 v2.15.3

---

06 **In-app subscription and billing management** NEW 75

Subscribe to Plus or Pro directly from `Settings → Agents` using the desktop account, with checkout and subscription management opening on Stripe without a separate website sign-in. Shows subscription status and paid-through dates, refreshes Agent credits after payment is confirmed, and surfaces Plans and billing from the sidebar account menu. Adds in-app subscription management via `Settings → Agents → Default → Plans and billing`, showing plan availability and refreshing agent credit balance on return to the desktop window. Shows a one-time invitation to sign in for free Default Agent credits, dismissing persistently across launches after the user signs in or selects `Not now`.

Subscribe to a paid plan or manage billing without leaving the desktop app.

> 📍 In the desktop app, go to Settings → Agents, then select your plan (Plus or Pro) to open Stripe checkout directly.

Check remaining Agent credits and subscription renewal date after a payment.

> 📍 Open the sidebar account menu, select Plans and billing to view your paid-through date and refreshed Agent credit balance.

View your plan, manage your subscription, and see refreshed credit balance (free + subscribed) without leaving the app.

> 📍 Settings → Agents → Default → Plans and billing

— Exact settings path and billing flow with working examples. v2.14.0 v2.13.0

---

07 **Persona library, gallery, and writing-role precedence** NEW 70

Adds a shared, editable Persona library with per-Chat Persona selection; new Chats inherit the project's last choice and reopened Chats restore their own Persona. Adds a browsable gallery of ready-made Personas from the composer, with sample reply comparison and the ability to add independent copies to your library. Gives native

Agents StashBase's writing role before the chat's persona. Adds a Builder persona for build-in-public updates.

Assign a saved Persona to a new Chat so the agent consistently adopts a specific tone or role across turns.

> 📍 In the Chat composer, open the Persona library, select a Persona, and start a new Chat — it will inherit your project's last Persona choice automatically.

Evaluate a ready-made Persona before committing to it by comparing its sample replies alongside your own library entries.

> 📍 In the Chat composer, open the Persona gallery, browse ready-made Personas, read their sample replies, then choose `Add to library` to save an independent copy.

— Good UI steps for library and gallery; persona precedence stays thin. `v2.15.2` `v2.16.11` `v2.13.1`

---

08 **Agent approval modes: Auto, Ask, Edit, Plan** NEW 70

Sets Auto mode as the new default, routing routine project edits directly while a bounded review handles other routine actions and prompts when uncertain or risky. Extends Default Agent to honor Ask, Edit, and Plan modes between turns: Edit approves ordinary project document writes; Plan rejects writes, commands, and delegated work.

Restrict the Default Agent so it can edit project documents freely but must seek approval before running commands or delegating work.

> 📍 In the Default Agent settings, set the approval mode to `Edit` — document writes are auto-approved while commands and broader actions still require explicit approval.

— Named modes with concrete approval behavior and settings path. `v2.16.0` `v2.15.2`

---

09 **Desktop usage analytics** NEW 70

Adds desktop usage analytics covering foreground activity, document engagement, and Agent turns, with retention reporting and pre-login activity linked to the signed-in account; the Settings toggle clears queued events and local analytics identifiers when disabled.

Disable analytics collection and clear all queued events and local analytics identifiers from the desktop client.

> 📍 In StashBase, go to Settings and toggle off the analytics/usage statistics option.

— Clear toggle location and scope of collected data. `v2.15.5`

---

10 **GitHub import via HTTPS snapshot without Git** IMPROVED

65

Gallery copies and GitHub imports now download a public repository's default-branch file snapshot over HTTPS, eliminating the requirement to have Git installed locally. Downloaded archives are bounded and validated — invalid paths, unsafe links, and malformed archives are refused. Git LFS and submodules remain unsupported in GitHub imports — projects relying on either will not be fully downloaded. Intel native search/extraction components require macOS 15; older Intel systems retain only project entry and editing functionality.

— Mechanism and limits well specified; no manual workaround given. v2.15.4

11 **Native installers for macOS, Windows, and Linux** NEW

60

Delivers separate Apple Silicon (arm64) and Intel (x64) macOS installers, signed with Developer ID and notarized by Apple. Adds Windows NSIS installer and Linux AppImage and deb packages.

— Named installer formats, but no download steps given. v2.16.11

12 **App update checking and installation flow** IMPROVED 60

Downloads updates in the background and waits for an explicit Restart to update action in the sidebar or Settings before installing and relaunching. Checks for desktop updates every 15 minutes and on app foreground or computer wake, without reopening StashBase; manual update checks remain available at any time.

— Concrete 15-minute interval and control name, no manual-check path. v2.16.8 v2.15.5

13 **Next-chat startup speed improvement** IMPROVED 60

Reduces next-chat runtime readiness from 1.1–2.8 seconds to 14–16 ms via a single prepared native runtime that eliminates repeated process/MCP setup between chats.

— Precise before/after numbers but nothing for the user to act on. v2.16.0

14 **Writing workspace appearance customization** NEW 60

Adds appearance themes, system font selection, reading layout, and interface text sizing to customize the writing workspace. Adds focus mode and typewriter scrolling as optional writing aids. Adds word count display with support for CJK text. Restores system spelling support without altering document content. Extends the Markdown reading bar with font and text-size controls, with persistent Serif/Sans and size preferences across sessions.

Switch to a Serif font at a larger text size for comfortable long-form document reading — preferences persist across sessions.

📍 In the Markdown reading bar, click the font control and select `Serif`, then use the text-size control to increase the size to your preference.

— Lists customization options with an example but no full settings path. v2.15.0 v2.12.0

THINNER COVERAGE BELOW

15 **Project window management** NEW 55

Reopens projects from the previous run in their own windows with saved document tabs, and adds a sidebar project switcher to select a recent project or open, create, or import one from GitHub, with each project opening in its own window or focusing its existing window without disturbing the current workspace.

Switch between multiple active projects without losing your current workspace — each project opens in its own window.

📍 In the sidebar, click the project name to open the project switcher, then choose a recent project or select `Open`, `Create`, or import from GitHub.

— Thin description with no configuration detail. Clear UI path to switch projects, no advanced options detailed.

v2.16.11 v2.16.2

16 **Native installers for Claude Code and Codex agents** NEW 55

Claude Code and Codex agents now install and update via their official native installers, with detection of existing installations, retry recovery from partial failures, and priority given to native launchers over stale npm entries.

— Mechanism named but no manual trigger steps. v2.15.3

17 **Passage-citing Chat from document selection** NEW 55

Supports sending selected document passages to the Agent, citing passages with links back to their source, and suggesting the open document in Chat.

Cite a specific passage from the open document in Chat so the Agent can reference it with a link back to its source.

📍 Select a passage in the open document, then choose `Send to Agent` from the selection toolbar; the Agent will cite the passage with a link back to its source and suggest the open document in Chat.

— Clear selection-to-chat workflow demonstrated in example. v2.12.0

18 **Automatic file refresh and conflict handling** IMPROVED

50

Open documents now automatically refresh external file changes, including directly referenced local JavaScript and CSS in HTML previews, with unsaved-draft conflict review available. Project files also refresh automatically when returning to the window, and folder switches retire stale refresh failures without replacing open drafts.

— Behavior described but no manual trigger or settings path. v2.15.3 v2.16.17

19 **Daily scheduled project tasks** NEW

50

Adds daily project tasks with a chosen agent, clear run outcomes, and missed-run history — missed tasks do not run late, and completed run chats follow notification settings.

— Mechanism described, but no setup steps shown. v2.16.15

20 **Antigravity native Agent runtime** NEW

50

Adds Antigravity native Agent runtime with streamed turns, tool approval, and conversation restoration.

Approve or deny agent tool calls in real time during an Antigravity-powered chat session.

> 📍 In the stashbase UI, open a chat, select the Antigravity (agy) runtime from the model selector, then respond to each tool-use prompt in the tool approval dialog as it appears during the streamed turn.

— Thin changelog text, but example gives exact runtime selection steps. v2.16.8

21 **Intel macOS 12–14 native search/extraction limitation** BREAKING

50

On Intel macOS 12–14, native search/extraction components are not available; only project entry and editing remain functional.

— Exact OS range named but no workaround beyond limited mode. v2.16.9

22 **Agent and MCP tool capabilities** NEW

45

Agent and MCP text tools now support exact-path UTF-8 code and configuration files, including dotfiles and extensionless files, while preserving binary-file protection and project permissions, and allow explicitly requested external reads and searches within the requested scope and runtime permissions.

— Clear scope but no example of invoking the tool. Single vague sentence with no mechanism or example.

v2.16.17 v2.15.1

23 **Agent runtime update prompts in model picker** IMPROVED

45

Offers an Agent update in the model picker only when a newer runtime release is known, with the manual updater still available in Settings. Adds runtime updates to the model picker that refresh the model catalog without sending the current draft.

– States behavior but no exact command or trigger path. v2.13.2 v2.16.11

24 **Markdown selection toolbar: Heading menu and narrow-pane simplification**

IMPROVED

45

Adds a Heading menu (Text and Heading 1–3) to the selection toolbar, keeping Humanize and Ask Agent visible in split panes. Simplifies the Markdown selection toolbar in narrow document panes while retaining Heading controls and Ask Agent.

– Describes toolbar changes without navigation steps. v2.12.0 v2.13.1

25 **'Update Codex' prompt for metadata/model conflicts** IMPROVED

45

Offers Update Codex when the same model produces both a native metadata warning and a ChatGPT model rejection, while preserving the original error.

– Specific trigger condition named but no resolution steps. v2.13.2

26 **Desktop agent notifications with system-settings recovery** NEW

30

Adds desktop agent notifications with system-settings recovery.

– Bare feature name with no detail on recovery mechanism. v2.16.10

27 **Collapsible completed-turn file lists in chat interface** NEW

30

Adds collapsible completed-turn file lists in the chat interface.

– One-line UI addition without interaction detail. v2.16.10

28 **Humanize feature retired** BREAKING

25

The Humanize feature has been retired and is no longer available.

– Bare removal notice with no migration guidance. v2.13.1

29 **Keyboard sidebar resizing** NEW

25

Supports keyboard sidebar resizing.

– Bare one-line addition without keys or shortcuts named. v2.13.2

30 **Editor active-line highlight only while focused** IMPROVED

20

Shows the editor active-line highlight only while the editor has focus.

— Minimal behavioral note with no further detail. v2.13.2

└──▷ BREAKING ON UPGRADE

- ! On Intel macOS 12–14, native search/extraction components are not available; only project entry and editing remain functional.
- ! Git LFS and submodules remain unsupported in GitHub imports — projects relying on either will not be fully downloaded.
- ! Intel native search/extraction components require macOS 15; older Intel systems retain only project entry and editing functionality.
- ! The `suggest_edits` tool is removed; external MCP clients must replace calls to it with `edit_file` (to edit) or `write_file` (to create documents).
- ! The Humanize feature has been retired and is no longer available.

WAS THIS USEFUL? 👍 👎

zvec

Feedback ⓘ v0.8.0 NOTES ↗ RANK

A lightweight, lightning-fast, in-process vector database

> Zvec v0.8.0 ships a major performance suite — uniform UINT4 quantization, PQ-INT4 FastScan, OPQ rotation, a shared page cache, and Turbo SIMD acceleration — alongside broader platform support (Windows, Android, iOS, Windows ARM64) and a breaking change to Jieba FTS dictionary configuration that removes several Python, C, and C++ APIs.

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

01 Configurable Jieba dictionary paths for FTS

BREAKING

90

Chinese full-text search now reads dictionary paths from `FtsIndexParam.extra_params.jieba_dict_dir` or the `ZVEC_JIEBA_DICT_DIR` environment variable, with `user_dict_path` to add business vocabulary, removing the need for separate dictionary files or global path configuration. This replaces the removed Python init(jieba_dict_dir=...), set_default_jieba_dict_dir(), and get_default_jieba_dict_dir() calls, plus the corresponding C configuration accessors, C default-directory APIs, and C++ global configuration field and registration APIs.

Override the embedded Jieba dictionary with a domain-specific one and add business vocabulary for Chinese FTS, without any global path setup.

python

```python
fts_params = zvec.FtsIndexParam(
    extra_params={"jieba_dict_dir": "/data/dicts/custom_jieba"},
    user_dict_path="/data/dicts/business_terms.txt"
)
```

— Exact config keys, env var, and removed APIs given; runnable example provided. v0.8.0

02 UINT4 quantization, PQ-INT4 FastScan, and OPQ

NEW

77

Adds uniform UINT4 quantization and integrates uniform UINT4/UINT7/UINT8 quantizers with the database layer and HNSW for a shared quantization range that avoids per-datapoint score reconstruction during queries. Adds PQ-INT4 and a 4-bit PQ FastScan implementation for IVF with packed block scanning, SIMD and scalar distance kernels, and precomputed residual distance tables, plus Optimized Product Quantization (OPQ) rotation preprocessing added to the Turbo framework.

— Mechanism detailed but no API or flag to invoke it directly. v0.8.0

03 **Broader platform and architecture support** NEW 75

Extends DiskANN to Windows via asynchronous unbuffered reads through I/O completion ports and to Android and iOS via portable `pread` support; adds macOS x86_64 (Intel Mac) and Windows ARM64 builds, including Windows ARM64 SDK packages and native Python 3.11–3.14 wheels; and adds Windows x86_64 support for RaBitQ indexes.

— Names exact platforms, Python wheel versions, and build artifacts. v0.8.0

04 **Faster refinement with query_internal_ids and FP16/UINT8** NEW 70

Adds Collection.query_internal_ids() to reduce ID lookup overhead during refinement for read-only collections; HNSW and Vamana can now refine using FP16 or UINT8 vectors, with `scale_factor` controlling the candidate budget.

— Named method and tuning param, brief mechanism, no example. v0.8.0

05 **Turbo SIMD acceleration for distance computation** IMPROVED 70

Accelerates distance computation across Flat, HNSW, IVF, and DiskANN by migrating to Turbo with expanded AVX2, AVX512, AVX512-FP16, SSE, and ARM NEON kernels and runtime dispatch, with major gains for INT4 kernels and Linux ARM64 workloads.

— Names kernels and index types but no tunable flag. v0.8.0

06 **Index build memory and performance optimizations** IMPROVED 70

Reduces peak memory during IVF and DiskANN index builds by streaming merged inputs, bounding temporary encoding buffers, and releasing completed builders earlier; accelerates HNSW RaBitQ construction by reusing raw vectors during neighbor pruning and batching Flat provider lookups and segment locking; and enables two-pass Vamana graph construction by default, optimizing candidate state, prefetching, and medoid selection for improved graph quality and search performance.

— Detailed mechanism per index type but no configurable surface. v0.8.0

07 **Shared page cache for vector indexes** NEW 65

Adds a shared page cache for vector indexes, integrating HNSW, IVF, and DiskANN with buffer-backed storage supporting pinned reads, batched page access, and DiskANN direct-I/O fallback under cache pressure.

— Mechanism and scope given but no configuration surface named. v0.8.0

08 **Expanded UTF-8 length limits for IDs and names** IMPROVED 65

Document ID support expands to valid UTF-8 up to 1,024 bytes; collection names now accept valid UTF-8 from 1 to 256 bytes; field names expand to 64 bytes.

— Exact byte limits given but no code example or key. v0.8.0

└──▷ **BREAKING ON UPGRADE**

! Python users must remove init(jieba_dict_dir=...) calls and all calls to set_default_jieba_dict_dir() / get_default_jieba_dict_dir() — these APIs are removed; use `FtsIndexParam.extra_params.jieba_dict_dir` or the `ZVEC_JIEBA_DICT_DIR` environment variable instead. The corresponding C configuration accessors, C default-directory APIs, and C++ global configuration field and registration APIs are also removed.

WAS THIS USEFUL? 👍 👎

---

ai-microcore

Feedback ⓘ v6.9.0 NOTES ↗ RANK

A handy lib for smooth interaction with large language models (LLMs) and crafting AI apps.

> ai-microcore v6.9.0 adds Python 3.14 support and introduces the mistral-medium-3-5 mid-tier model preset.

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

01 **Python 3.14 support** IMPROVED 50

The library now supports Python 3.14, extending the tested version range to 3.10–3.14 with full CI coverage across all backends.

— Names version range and CI scope but no code example v6.9.0

02 **mistral-medium-3-5 model preset** NEW 30

Adds the `mistral-medium-3-5` mid-tier model preset.

— Named preset but no usage detail or mechanism v6.9.0

WAS THIS USEFUL? 👍 👎

synthadoc

Feedback ⓘ v1.4.3 NOTES ↗ RANK

An open-source LLM knowledge compilation engine that turns raw documents into structured, local-first wikis, offering a transparent alternative to traditional RAG.

> Synthadoc's biggest window ships its 8th agentic maintenance workflow (Content Health Resolver), an official multi-arch Docker image with compose and CI examples, expanded cross-wiki fan-out querying with explicit routing control, an upgraded OKF v0.2 export format, and Anthropic prompt caching that cuts ingest costs 80-90%.

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

01 **Content Health Resolver workflow and config** NEW 95

Adds the Content Health Resolver, the 8th agentic maintenance workflow: a heuristic scanner that detects six issue types — `duplicate_section`, `empty_section`, `thin_section`, `unclosed_code_block`, `orphaned_heading`, and `broken_list_nesting` — without LLM cost, then proposes LLM-generated rewrites with side-by-side BEFORE/AFTER diff approval and one-step rollback via lifecycle snapshot. Run it with `synthadoc workflow run --name content-health` (all flagged pages) or `--slug <slug>` (one page), view history with `list`, and undo with `roll back <slug>`; it's also exposed in the web UI as a pre-prompt hint chip (Run content health resolver), processing pages highest-severity first with per-page Approve/Skip controls. The scanner's sensitivity is configurable via the `[content_health]` section in `config.toml`, with `thin_section_min_words` uncommented and defaulting to 40.

Scan all flagged wiki pages for structural issues and approve or skip each proposed fix interactively — no LLM cost until you start the workflow.

```
$ synthadoc workflow run --name content-health
```

Target a single page (e.g. alan-turing) for content health repair, then roll it back if the result is unsatisfactory.

```
$ synthadoc workflow run --name content-health --slug alan-
turing
# later, if needed:
synthadoc workflow run --name content-health roll back alan-
turing
```

Set the minimum word count for thin-section lint findings in your wiki config.

```toml
[content_health]
thin_section_min_words = 40
```

— Full command set, issue list, config key, and runnable examples given. v1.4.2 v1.4.3

02 **OKF v0.2 wiki export format** BREAKING 95

`synthadoc export --format okf --output <path>` exports a wiki as an OKF v0.2 bundle (upgraded from v0.1) — directories of Markdown with typed YAML frontmatter including `type`, `description`, `status` (normalised to `draft` / `stable` / `deprecated`), a `generated: {by, at}` block (replacing `timestamp`), `tags`, `synthadoc_lifecycle`, and optional `sources`, `verified`, `stale_after` fields; `index.md` frontmatter is simplified to `okf_version: '0.2'` only. Bundles are consumable by any OKF-aware agent without a running Synthadoc instance, and also exportable via the Obsidian plugin through Command Palette → Synthadoc: Export Wiki → OKF v0.2 bundle. Breaking: the `timestamp` field is removed in favour of `generated.at`; `status` now uses OKF spec values instead of Synthadoc lifecycle values; `index.md` frontmatter drops `type`, `title`, `description`, and `timestamp`; and `log.md` is now plain Markdown with no frontmatter.

— Exact command, frontmatter fields, and UI path all named. v1.4.0

03 **Cross-wiki fan-out querying and routing** IMPROVED 90

`synthadoc query --cross-wiki` fans a question out to all registered running wikis in parallel and returns a unified answer with `[[wiki-name::PageTitle]]` citation pills, skipping offline peers with an amber warning; `synthadoc serve --all --background` starts every registered wiki server in one command. `synthadoc cross-wiki routing init` and `synthadoc cross-wiki routing edit` generate and manage an explicit per-topic routing table that overrides LLM auto-routing, with the lint agent now enforcing routing rules as lint warnings before a query runs. Also adds automatic translation of CJK (Chinese, Japanese, Korean) questions into the target wiki's language before fan-out, per-peer epoch cache invalidation, and saving of cross-wiki query sessions to Recent Runs in the web UI.

— Commands and UI location named but no runnable example given. v1.4.0

04 **Official Docker image and deployment artifacts** NEW 90

Ships an official OCI image `chenp/synthadoc` for `linux/amd64` and `linux/arm64`, runnable on Docker Engine, Docker Desktop, Apple `container` (macOS 26+, Apple Silicon), and WSL Containers (`wslc`, Windows 11). Adds `docker/compose/multi-wiki.yml` for running multiple domain wikis with an inlined cross-wiki registry,

`docker/compose/with-ollama.yml` for fully local API-key-free deployments with an Ollama sidecar, and `docker/examples/github-actions-ingest.yml` for nightly ingest and lint jobs in CI. The new `synthadoc serve --host` flag binds the server to a specific host interface for Docker and shared-host deployments.

Run a shared team wiki in Docker, pinned to this release, with the wiki folder mounted from the host and the API key supplied via an env file.

```
$ docker run -d --name my-wiki -p 7070:7070 --env-file .env -v
~/wikis/my-wiki:/wiki chenp/synthadoc:1.4.2
```

— Named image, compose files, platforms, and a runnable docker run command. v1.4.2

05 **In-container plugin and template management** NEW 80

`synthadoc plugin upgrade` and `synthadoc templates sync` (including `--force`) can now run directly inside a running Docker container via `docker exec`, removing the need for a host-side Python installation.

Push plugin and template updates to a containerised Synthadoc instance without installing Python on the host.

```
$ docker exec <container> synthadoc plugin upgrade
docker exec <container> synthadoc templates sync --force
```

— Exact commands and runnable docker exec example given. v1.4.3

06 **DashScope/Qwen provider support** NEW 70

Adds `DASHSCOPE_API_KEY` as an accepted alias for `QWEN_API_KEY`, plus `QWEN_BASE_URL` to override the DashScope endpoint, alongside two new free-tier models on the DashScope provider: `qwen3.8-flash` ($0.15/$0.47 per 1M tokens) and `qwen3.8-max` ($2/$6 per 1M tokens), both confirmed working with `thinking=disabled`.

— Named env vars and model pricing but no usage example. v1.4.3

07 **Anthropic prompt caching for ingest** NEW 70

Activates Anthropic prompt caching by sending `cache_control: {"type": "ephemeral"}` on every Anthropic API call, with cache token counts (`cache_read_input_tokens` and `cache_creation_input_tokens`) flowing through to estimate_cost(). No configuration change is needed, and it delivers 80-90% ingest cost savings on repeated queries within the 5-minute cache window.

— Clear mechanism and numbers but automatic, no command to run. v1.4.3

THINNER COVERAGE BELOW

08 Template sync removes superseded demo files IMPROVED

40

`synthadoc templates sync` now automatically removes superseded binary demo source files (same stem, different extension) on upgrade, replacing `.pptx`, `.xlsx`, and `.pdf` demo sources with plain-text equivalents.

— Behaviour named but thin, no example or further detail. v1.4.3

└──▷ BREAKING ON UPGRADE

! The `timestamp` field is removed from OKF export output and replaced by `generated.at`; any downstream consumer reading `timestamp` from an OKF bundle will no longer find it.

! The `status` field in OKF export now uses OKF spec values (`draft` / `stable` / `deprecated`) instead of Synthadoc lifecycle values (`active`, `draft`, …); consumers that matched on the old values must update their logic.

! `index.md` frontmatter in OKF bundles is reduced to `okf_version: '0.2'` only — the `type`, `title`, `description`, and `timestamp` fields previously present are removed.

! `log.md` in OKF bundles is now plain Markdown with no frontmatter; consumers that previously read `type`, `title`, or `timestamp` from `log.md` frontmatter will find none.

WAS THIS USEFUL? 👍 👎

askimo

Feedback ⓘ v1.5.4 NOTES ↗ RANK

AI desktop app for chat, RAG, Skills, MCP tools, and agents. Support multiple LLMs (Anthropic, OpenAI, VertexAI, vLLM, Nvidia NIM, Gemini, OpenRouter)

> Askimo v1.5.4 adds support for running local LLM engines alongside a broader voice stack built on `whispercpp` and `SystemTextToSpeech`, plus improvements to MCP observability, inline autocomplete, and the LLM Browser.

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

01 Voice and speech processing support NEW 67

Added `whispercpp` integration for voice processing and model management, supporting bidirectional voice communication; added a `SystemTextToSpeech` provider, broadening text-to-speech output modality options; added live transcription support in voice stream handling with an associated UI toggle; and added Whisper model download cancellation support for large model asset retrieval.

Enable live transcription during a voice session to capture spoken input as real-time text in the chat interface.

> 📍 In the chat interface, start a voice session, then activate the live transcription toggle in the voice stream controls.

– Names whispercpp and SystemTextToSpeech but lacks config/flag detail v1.5.4

THINNER COVERAGE BELOW

02 LLM Browser model discovery and i18n IMPROVED 35

Expands LLM Browser with improved model discovery and internationalization (i18n) support.

– Names the LLM Browser area but no specific locales or discovery details v1.5.4

03 Inline autocomplete typing suggestions NEW 33

Adds inline autocomplete typing suggestions with refined suggestion matching logic and usage history management.

– Mentions matching logic and history but no concrete usage path v1.5.4

04 MCP instance tracking and broadcasting NEW 30

Adds MCP (Model Control Plane) instance tracking and change broadcasting for enhanced system observability.

– Named MCP but no concrete endpoint or UI described v1.5.4

05 **Local LLM engine integration** NEW 26

Adds local LLM engine integration support for running external Large Language Models locally, increasing deployment flexibility.

— States the capability but no mechanism or named engines v1.5.4

WAS THIS USEFUL? 👍 👎

matrixone

Feedback ⓘ v4.2.6 NOTES ↗ RANK

AI-native HTAP database with Git-for-Data and built-in vector search, serving as the data and memory backbone for intelligent agents and applications.

> MatrixOne v4.2.6 broadens vector-search SQL to cover equi-joins and scalar subqueries, trims memory use in median aggregation, and requires a DECIMAL256 index rebuild for affected deployments.

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

01 **DECIMAL256 index rebuild required after upgrade** BREAKING 70

Existing DECIMAL256 indexes affected by legacy writes must be rebuilt from their base tables in a maintenance window before user reads and writes resume; mixed old-and-new writer deployments are not supported after upgrading.

– Clear migration requirement but no rebuild command given. v4.2.6

02 **Expanded SQL support for vector index queries** NEW 62

IVF-FLAT Top-K queries can now run through eligible equi-joins, enabling vector similarity search inside join patterns, and both IVF-FLAT and HNSW indexes now accept query vectors supplied via scalar subqueries, broadening the SQL shapes usable for nearest-neighbor lookups.

– Names index types and SQL shapes but no example query syntax. v4.2.6

THINNER COVERAGE BELOW

03 **Reduced memory amplification in median aggregation** IMPROVED 28

Median aggregation now reduces intermediate-state memory amplification, lowering peak memory consumption for large aggregations.

– States the improvement but no mechanism or measured impact. v4.2.6

└──▷ **BREAKING ON UPGRADE**

! Existing DECIMAL256 indexes affected by legacy writes must be rebuilt from their base tables in a maintenance window before user reads and writes resume; mixed old-and-new writer deployments are not supported after upgrading.

WAS THIS USEFUL? 👍 👎

Crow Desktop

Feedback ⓘ v0.1.2 NOTES ↗ RANK

Crow: build your second brain from the documents on your computer. Downloads for macOS, Windows and Linux.

> Crow Desktop launched as a cross-platform app (macOS, Windows, Linux) for indexing local documents into a searchable knowledge base, then quickly added note provenance tracking, deletion, and reset controls, plus an upgrade to native Office/OpenDocument reading that drops support for `.odp` files.

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

01 **Reset Crow wipes wiki and settings** NEW 80

Adds Settings → Library → Reset Crow to erase the wiki, chats, and settings and restart onboarding, while leaving source documents intact and giving the option to stay signed in to OpenRouter.

– Exact settings path and precise scope of what is erased v0.1.1

02 **Document format support, CSV and native Office reading** BREAKING 75

At launch Crow read Word, PowerPoint, Excel, and OpenDocument files as text alongside PDF, Markdown, and plain text. A later release added support for reading CSV files as plain text, and upgraded to llmwiki2 0.5.0 for native reading of Word, PowerPoint, Excel, and OpenDocument files, replacing Crow's own converter and filing spreadsheets as one note per sheet. As a result, OpenDocument presentation files (`.odp`) are no longer read, since llmwiki2 0.5.0 does not support them, so previously ingested `.odp` files will no longer be processed.

– Names the library upgrade and breaking format loss but no migration steps v0.1.2

v0.1.0

03 **OpenRouter sign-in via OAuth or API key** NEW 75

Adds an OpenRouter OAuth (PKCE) browser sign-in that provisions a per-app key on the user's account, with Settings able to sign out and open the key's page on openrouter.ai. Supports `OPENROUTER_API_KEY` from the environment or a `.env` file as an alternative to the browser flow.

– Names env var and auth flow but no setup walkthrough v0.1.0

04 **Note provenance with source file access** NEW 65

Notes now list the documents they were filed from, with options to open the original, reveal it in Finder/Explorer, or fall back to Crow's stored text copy if the file was moved or deleted.

— Explains the fallback behaviour but not an exact UI path v0.1.1

05 **Privacy notice on model-selection screens** NEW 65

Adds a privacy notice at model-selection screens (Connect and Settings → Model) disclosing that privacy depends on OpenRouter and the chosen provider, with links to each provider's data policy, and describes exactly what the model reads: each document's text, name, and location, plus the notes and earlier messages relevant to each question.

— Names exact screens and disclosed data scope v0.1.1

06 **Model selection including Advanced settings** NEW 60

Adds model selection from Gemini 3.8 Flash, Claude Sonnet 5.5, or DeepSeek V4.1 Flash, plus any OpenRouter model under Advanced settings.

— Names specific models and settings location, no mechanism detail v0.1.0

07 **Cross-platform installers with self-updating** NEW 60

Ships installers for macOS (Apple silicon and Intel), Windows, and Linux with self-updating from the crow-hq/crow-desktop GitHub releases.

— Names platforms and update source, a clear download starting point v0.1.0

08 **Delete and reveal-in-file-manager actions in the brain** NEW 60

Adds the ability to delete a note or an entire folder (with all its contents) from the open note view or via right-click in the brain, without touching the source documents, and adds Show in Finder/Explorer/file manager for the wiki, a folder, or a note via right-click in the brain or the button on the open note.

— Names both right-click actions and their entry points v0.1.1

THINNER COVERAGE BELOW

09 **Workspace with chat, brain view and source highlighting** NEW 55

Adds a workspace with chat, a persistent brain view, source highlighting, context-aware follow-up questions, and drag-and-drop filing into brain folders.

— Lists the sub-features but gives no navigation or usage detail v0.1.0

10 **Five-screen onboarding with live folder counting** NEW

55

Adds a five-screen onboarding that lets users pick folders, date ranges, or single files, with live home-folder counting without opening or downloading cloud-only files.

— Describes the flow concretely but no exact screen names v0.1.0

11 **Live cost and time estimate before indexing** NEW

45

Shows live cost, computed from OpenRouter's live prices and actual document sizes, and estimated time before indexing begins.

— Describes the computation basis but no numbers or UI path v0.1.0

└──▷ BREAKING ON UPGRADE

! OpenDocument presentation files (`.odp`) are no longer read: llmwiki2 0.5.0 does not support them, so any `.odp` files previously ingested will no longer be processed.

WAS THIS USEFUL? 👍 👎

uteke

Feedback ⓘ v0.21.1 NOTES ↗ RANK

The Brain for Your AI — Local-first memory engine for AI agents. Store, recall, and search memories with semantic embeddings.

> Over four releases uteke hardened its software-supply-chain with minisign- and cosign-verified upgrades, added per-tool MCP access gates for read-only tokens, shipped budgeted context packing, a knowledge-graph endpoint, verify/repair and timeline tools, and tightened server security with CORS defaults, body limits, and atomic document writes.

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

01 Signature verification and hardening for `uteke upgrade`

85

BREAKING

`uteke upgrade` now verifies minisign signatures over `checksums-sha256.txt` using the embedded public key `scripts/uteke-release.pub`, checking the trusted comment `uteke <tag>` and `checksums-sha256.txt.minisig` before trusting any hash — a missing or invalid signature is a hard failure (set `UTEKE_UPGRADE_SKIP_CHECKSUM=1` to bypass). It also restricts installs to stable `vX.Y.Z` tags, refuses downgrades, and rejects pre-release/build-metadata tags; the release tag is validated against `[0-9A-Za-z.-]`, the archive is capped at 128 MiB and `checksums-sha256.txt` at 1 MiB while streaming, and archive entries must be plain relative paths. `checksums-sha256.txt` is additionally signed via cosign keyless signing and published as `checksums-sha256.txt.bundle`, with the certificate bound to the repository workflow identity.

— Mechanism and flags named; no CLI invocation example given v0.21.1 v0.21.0

02 Knowledge graph endpoint `GET /graph`

85

BREAKING

Adds `GET /graph?node_id=&limit=` and MCP tool `uteke_graph` to list the latest edges with a configurable `limit`; the response includes `node_id` / `limit` query parameters, a `truncated` flag, and edges carrying `source` / `target`. Graph node ids are now memory ids instead of graph-node ids, and edges now come from live `memory_edges` plus explicit graph edges — clients parsing the previous response shape will break.

Query the knowledge graph for edges connected to a specific memory node, limiting results to avoid truncation.

```
$ curl 'http://localhost:8767/graph?node_id=0192abcd-1234-7def-abcd-0123456789ab&limit=50'
```

— Full query params and runnable example; shape-break noted v0.20.0

03 **Budgeted context packing via `--pack`** NEW 85

Adds `uteke recall --pack [--budget <chars>] [--exclude-ids <ids>]` for character-budgeted context packing, returning a `{selected, skipped, budget_used, budget_chars}` envelope with per-item skip reasons (`excluded`, `budget`) instead of a bare ranked list. Also surfaced via `POST /recall` (`pack`, `budget_chars`, `exclude_ids`) and MCP `uteke_recall` (`pack`, `budget_chars`, `exclude_ids`).

Inject only the memories that fit a 4 000-character prompt budget, skipping IDs already in the current context window.

```
$ uteke recall 'deployment steps' --pack --budget 4000 --exclude-ids id1,id2
```

— Flags, envelope shape and runnable example all given v0.19.0

04 **Project-local config can no longer set credentials** BREAKING 80

Project-local `.uteke/uteke.toml` can no longer set `embedding.{backend,api_key,base_url,endpoint_path}`, `embed_fallback.*`, `extraction.*` credentials/endpoints, or the `[server]` section; these must move to the global config, or `UTEKE_TRUST_PROJECT_CONFIG=1` must be set to trust the file.

Trust a project-local config file that sets embedding credentials, when the global config is not available in a CI environment.

```
$ UTEKE_TRUST_PROJECT_CONFIG=1 uteke-serve --port 8767
```

— Exact keys and env-var override with runnable example v0.20.0

05 **Room-focused MCP tools gain recall parity and filtering**

IMPROVED 75

`uteke_room_recall` gains parity with `uteke_recall` via optional `min_score`, `tags` (any-of), `pack` with `budget_chars` and `exclude_ids`, `full_ids`, per-result memory id, `skipped` reasons in pack mode, and `limit` applied after `exclude_ids`.

`uteke_room_list` gains an optional `name` filter for case-insensitive substring match on room id or title, combined with `namespace`.

— Every parameter named, no example provided v0.21.0

06 **API Route Registry** NEW 75

Adds `GET /routes` to `uteke-serve`, returning the full endpoint registry as JSON (method, path, tier CORE/LAB, description, request/response types, related issues) from `api_registry::ENDPOINTS`, now tier-tagged with 33 CORE and 44 LAB endpoints and a generated `docs/core-contract.json` consumed by uteke-cloud's route-parity gate, so CORE endpoint changes must go through the cloud contract process.

Fetch the full live API contract so a typed client or agent skill can validate its assumptions against the real server.

```
$ curl -s http://localhost:8767/routes | jq '[.[] | {method, path, tier}]'
```

— Endpoint and fields named with a runnable example Counts and artifact named, no reader action described

v0.19.0

07 **`POST /verify` and `POST /repair` for vector index** NEW 75

Adds `POST /verify` and `POST /repair` HTTP endpoints plus MCP tools `uteke_verify` and `uteke_repair` to audit and rebuild the vector index from stored embeddings without restarting `uteke-serve`.

Detect and fix a stale vector index (rows in SQLite but missing from HNSW) on a live server without restarting it.

```
$ curl -s -X POST http://localhost:8767/verify -H 'Authorization: Bearer <write-token>' | jq . && curl -s -X POST http://localhost:8767/repair -H 'Authorization: Bearer <write-token>'
```

— Endpoints named with a runnable live-repair example v0.19.0

08 **CLI logs moved to stderr** BREAKING 70

CLI log lines now go to stderr instead of stdout, so `--json` output stays parseable by json.loads() even when warnings or errors are emitted; scripts that previously read log lines from stdout must be updated to read stderr instead.

Pipe `--json` output safely to a parser in a script, now that warnings go to stderr and no longer corrupt stdout.

```
$ uteke recall 'deployment steps' --json 2>uteke.log | python3 -
c 'import sys,json; print(json.load(sys.stdin))'
```

— Clear before/after backed by a runnable example v0.21.1

09 **Separate recall_count tracking from access_count** BREAKING

70

Adds a `recall_count` field to Memory (schema v21) so recall and search hits are tracked separately from explicit `get` operations, preventing automated recalls from over-reinforcing memories in salience scoring. As a result `access_count` no longer increments on `recall`, `recall_context`, or `search` — a default (`fusion`) recall previously bumped the counter 3 times per memory and counted unreturned candidates, but counters are now touched once per user-level call on final results only, and historical counts cannot be split.

— Full before/after given, no migration command shown v0.21.0

10 **`uteke guide` command and `GET /guide` endpoint** NEW

70

Adds a `uteke guide` CLI command and `GET /guide` HTTP endpoint, plus a new `instructions` field in the MCP `initialize` response, pointing agents to workspace memory conventions before writing.

Ask a running uteke server to show the workspace memory conventions guide, so an agent knows where to look before writing new memories.

```
$ curl -s -H 'Authorization: Bearer <token>'
http://localhost:8767/guide
```

— Named endpoint paired with a runnable curl example v0.21.0

11 **Server now applies recall and tiering config keys** IMPROVED

70

`uteke-serve` now applies `[embed_fallback]`, `graph_density_weight`, `graph_authority_weight`, `graph_rerank_enabled`, `jaccard_weight` from `[recall]`, and `hot_days`, `warm_days`, `hot_boost` from `[tier]` in `uteke.toml`, which were previously read by the CLI only.

— Exact config keys named, no sample config shown v0.21.0

12 **Non-default namespaces enter deprecation (phase 1)** DEPRECATED

70

Writing to a non-default namespace now returns a `warning` field on HTTP `/remember`, `/room/remember`, `/extract`, `/import`, `/room/create`, `PUT /memory`, in MCP tool text for `uteke_remember` / `uteke_update` / `uteke_room_create`, and as a stderr line from `uteke remember` / `uteke import` / `uteke room create`.

— All surfaces named, no migration path given v0.21.0

---

13 **Server-side request hardening: CORS, body limits, Host header**

70

BREAKING

CORS is now off by default with wildcard `*` opt-in; 500 responses carry a generic body and request query strings are no longer logged. Adds `/import` size-limit enforcement, a `/mcp` body limit and CORS handling (an oversized `/mcp` body now returns `413`), and `resolve_id` hardening so ids accept only hex digits and dashes. Host header validation now blocks DNS rebinding.

— Every surface named, no config example to opt in v0.20.0

---

14 **Per-tool MCP access gates for read-only tokens** NEW

65

Each MCP tool is now gated individually: a read-only token can call the read tools (`recall`, `search`, `list`, `get`, `stats`, room and document reads) via `POST /mcp`, while every write tool returns `isError: true` for that token.

— Endpoint and tool names given, no setup steps shown v0.21.0

---

15 **`doc export` command for documents** NEW

65

Adds a `doc export` command that writes every document to `--output`, honouring the namespace; JSON export is an array of full documents.

Export all documents in a namespace to a JSON file for offline audit or backup.

```
$ uteke doc export --output ./backup-docs.json --namespace
project-alpha
```

— Flag and format named with a runnable example v0.20.0

16 **MCP `uteke_timeline` tool for memory history** NEW 65

Adds MCP tool `uteke_timeline` (params: `id`, optional `limit`) backed by a new `timeline_events` table (schema v20) that survives memory deletion and deprecation, recording `deprecated` and `forgot` tombstone events.

— Params and schema named, no example invocation v0.19.0

17 **Stricter id validation on `DELETE /forget`** BREAKING 60

`HTTP DELETE /forget` now rejects a non-hex id with 400 Invalid memory id '<id>' (expected a UUID or hex prefix), and requests using `%` or `_` as LIKE wildcards in the id parameter are blocked.

— Exact endpoint and error named, no example call shown v0.21.0

18 **Vector Index Integrity & Performance** IMPROVED 60

The `.keys` sidecar file now carries a digest and row count, with a mismatch or missing sidecar triggering an automatic index rebuild from SQLite on load; index saves are now batched (firing after 64 mutations or 2 seconds, with flushes on shutdown and drop) instead of rewriting the whole index on every write, and the vecq index is compacted automatically when dead rows exceed 25%.

— Mechanism described, nothing for reader to invoke Concrete thresholds given, nothing actionable to run

v0.20.0

19 **Atomic document upsert with cycle detection** BREAKING 60

Document upsert/update is now atomic — if embedding fails nothing is written, whereas previously a partial write could occur. `doc upsert` re-parenting moves the whole subtree and rejects cycles or hierarchies deeper than 10 levels.

— Before/after and limit given, no example call v0.20.0

20 **Detailed remember variants return write outcome** IMPROVED 60

Adds `remember_detailed`, `remember_typed_detailed`, and `remember_auto_infer_detailed` returning a `RememberOutcome { id, embedding_written, warning }` struct; `POST /remember` and `POST /room/remember` responses now carry `embedding_written` and `warning` fields.

— Struct and endpoint fields named, no usage example v0.19.0

THINNER COVERAGE BELOW

21 **Hard caps on list and extract result sizes** BREAKING

55

List `limit` is now capped at 1000 on `/list`, `/recent`, `/contradictions`, `/doc/list`, and deprecated listings, and `/extract` `max_facts` is capped at 100 — requests relying on higher limits now get truncated results.

— Exact endpoints and caps named, no workaround given v0.20.0

---

22 **Full UUID display option for MCP list/recall tools** NEW

50

Adds a `full_ids` option for MCP tools `uteke_list`, `uteke_recall`, and `uteke_room_memories` to print full UUIDs instead of the ambiguous 8-character prefix.

— Option and tools named, no example invocation v0.20.0

NEW INTEGRATION Hermes Desktop plugin package

└──▷ BREAKING ON UPGRADE

- ! `uteke upgrade` now fails closed if `checksums-sha256.txt.minisig` is absent or invalid — releases without a minisign signature cannot be installed; this behaviour is new from v0.21.1 onward (a v0.21.0 binary never verifies signatures).
- ! Scripts that read log lines from stdout must be updated to read stderr instead; `--json` output no longer includes log lines.
- ! `access_count` no longer grows on `recall`, `recall_context`, or `search` — those now increment the new `recall_count` field. Any tooling or queries that read `access_count` as a proxy for total memory usage will see lower values after the schema v21 migration.
- ! A default (`fusion`) recall previously bumped the counter 3 times per memory and counted unreturn candidates; counters are now touched once per user-level call on final results only. Historical counts cannot be split.
- ! `HTTP DELETE /forget` now rejects a non-hex id with 400 Invalid memory id `<id>` (expected a UUID or hex prefix); requests using `%` or `_` as LIKE wildcards in the id parameter are blocked.
- ! List `limit` is capped at 1000 on `/list`, `/recent`, `/contradictions`, `/doc/list`, and deprecated listings; `/extract` `max_facts` is capped at 100 — requests that relied on higher limits will now receive truncated results.
- ! CORS wildcard `*` is no longer on by default — integrations that depended on open CORS must explicitly opt in.
- ! Project-local `.uteke/uteke.toml` can no longer set `embedding.{backend,api_key,base_url,endpoint_path}`, `embed_fallback.*`, `extraction.*` credentials/endpoints, or the `[server]` section — these must move to the global config, or `UTEKE_TRUST_PROJECT_CONFIG=1` must be set.
- ! `GET /graph` node ids are now memory ids instead of graph-node ids; edges now come from live `memory_edges` plus explicit graph edges; edges carry `source` / `target` — clients parsing the previous response shape will break.

! Document upsert/update is now atomic — if embedding fails nothing is written, whereas previously a partial write could occur.

v0.19.1 NOTES ↗ STABLE

v0.19.1 hardens `uteke upgrade` version resolution (302-redirect probe, tag validation, `GITHUB_TOKEN` bearer support) and upgrade workspace security (private temp dir, timeouts, checksum filename matching).

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.19.1
https://github.com/codecoradev/uteke.git
# already have the repo? check out this version:
$ git checkout v0.19.1
```

Adds `integrations/hermes/` plugin package so Hermes Desktop detects the repo in its Install plugin dialog and catalog; template changes require re-rendering via `scripts/render-hermes-integration.sh` (byte-parity guard test).

WAS THIS USEFUL? 👍 👎

Tencent WeKnora

Feedback ⓘ v0.8.3 NOTES ↗ RANK

Open-source LLM knowledge platform: turn raw documents into a queryable RAG, an autonomous reasoning agent, and a self-maintaining Wiki.

> WeKnora v0.8.3 adds a cross-knowledge-base Wiki search endpoint, an invite-only auth tier, a new image gallery and admin model console, broader document-parsing and sync integrations, and unified reranking with retrieval diagnostics.

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

01 **Cross-knowledge-base Wiki search endpoint** NEW 70

Adds a new `POST /wiki-search` endpoint enabling search across multiple knowledge bases' Wikis rather than within a single one.

— Named endpoint and method given, but no request/response detail v0.8.3

THINNER COVERAGE BELOW

02 **Knowledge base sync and asset handling improvements** IMPROVED 55

Improves knowledge base integrations: adds Confluence directory sync, private image asset support, and preserves TOC folder hierarchy for Yuque synced documents.

— Three named integrations but no config or endpoint details v0.8.3

03 **Image gallery tab and viewer rework** NEW 50

Adds an image gallery tab with schema-driven filters, a reworked toolbar, header tabs and an image viewer.

— UI surfaces named but no exact navigation path v0.8.3

04 **MinerU 4.0 V1 API support** NEW 50

Supports the MinerU 4.0 V1 API for self-hosted MinerU document parsing.

— Names exact API version but no setup steps v0.8.3

05 **Cited passage navigation in chat** IMPROVED 40

Opens cited documents directly at the cited passage when clicked from chat responses.

— Clear UI behavior but no further mechanism detail v0.8.3

06 **Host-installed skills in Lite edition** NEW 40

Adds the ability to install and run skills on the host within the Lite edition.

— Names the edition and capability but no install steps v0.8.3

07 **Unified rerank and retrieval diagnostics** IMPROVED 35

Unifies rerank behavior across search APIs and exposes retrieval diagnostics.

— Vague on which APIs or what diagnostics are exposed v0.8.3

08 **Invite-only registration tier** NEW 30

Adds an invite-only registration tier to the auth system, alongside existing registration options.

— Bare mention, no mechanism or config shown v0.8.3

09 **System admin model catalog console** NEW 30

Adds a system admin console for managing a catalog of models.

— Only a name, no mechanism or location given v0.8.3

WAS THIS USEFUL? 👍 👎

dosco GraphJin

Feedback ⓘ v3.21.13 NOTES ↗ RANK

One governed graph for AI agents — GraphQL + MCP over your databases, files, APIs, and code

> GraphJin added JSON column query operations and agent catalog discovery powered by Ax decisions.

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

01 **Operations on keys inside JSON columns** NEW 40

GraphJin now supports query, order, page, aggregate, and merge operations on keys inside JSON columns.

— Names operation types but no syntax or example given v3.21.13

02 **Catalog discovery and skills preparation in agent** NEW 25

GraphJin introduces catalog discovery and skills integration in the agent, driven by Ax decisions.

— Vague description with no mechanism or usage detail v3.21.8

WAS THIS USEFUL? 👍 👎

vector-graph-rag

Feedback ⓘ v0.4.0 NOTES ↗ RANK

Graph RAG with pure vector search, achieving SOTA performance in multi-hop reasoning scenarios.

> vector-graph-rag's latest two releases switch the default LLM to GPT-6 Luna over the Responses interface with tunable reasoning effort, add two-stage Jev retrieval reranking and per-task model configuration, and introduce new retrieval evaluation baselines.

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

01 GPT-6 Luna default LLM via Responses interface

BREAKING

91

Switches the default generation model from GPT-4o-mini to GPT-6 Luna, used for triplet extraction, entity extraction, relationship reranking, and answer generation via the OpenAI Responses interface. Adds `VGRAG_LLM_REASONING_EFFORT` (defaults to `medium`) to tune reasoning depth for GPT-6 Responses requests, separates cache keys for the Responses interface and reasoning effort level, and rejects incomplete, refused, or empty Responses outputs instead of caching them as successful answers. Breaking: custom OpenAI-compatible endpoints must support the Responses interface, the OpenAI SDK minimum version is now 1.66.0, and existing GPT-4o-mini caches are not reused for GPT-6 calls — set `VGRAG_LLM_MODEL` explicitly to retain the previous model.

Pin reasoning effort to `low` for faster, cheaper generation while evaluating throughput on a large corpus.

```
$ VGRAG_LLM_REASONING_EFFORT=low python my_pipeline.py
```

Keep using GPT-4o-mini instead of the new GPT-6 Luna default so existing caches and custom endpoints remain valid.

```
$ VGRAG_LLM_MODEL=gpt-4o-mini python my_pipeline.py
```

Set model and reasoning effort persistently via a `.env` file so every run in the project shares the same configuration.

```ini
VGRAG_LLM_MODEL=gpt-4o-mini
VGRAG_LLM_REASONING_EFFORT=medium
```

— Names env var, SDK version floor and migration commands v0.4.0

02 **Two-stage Jev retrieval reranking** NEW 83

Setting `reranker_model="jev"` enables two-stage Jev retrieval, which selects graph relations and reranks their source passages alongside direct vector-search candidates. Explicit Jev versions and the `jev-latest` / `jev-preview` aliases are supported with no additional installation extra.

Use two-stage Jev reranking to improve multi-hop recall without installing any extra dependencies.

```python
from vector_graph_rag import VectorGraphRAG

rag = VectorGraphRAG(
    llm_model="gpt-4o",
    reranker_model="jev-latest",
)

result = rag.query("Who influenced Einstein's theory of relativity?")
print(result.answer)
```

— Exact parameter and runnable code example given v0.3.0

03 **Per-task model configuration via ModelConfig** NEW 68

Adds `ModelConfig` to configure `extractor_model`, `reranker_model`, and `answer_model` independently with task-specific endpoints and credentials, using `llm_model` as the shared default.

— Names config fields but no usage example v0.3.0

THINNER COVERAGE BELOW

04 **GPT-6 Luna and DeepSeek V4.1 Flash eval baselines** NEW 40

Adds GPT-6 Luna and DeepSeek V4.1 Flash as retrieval evaluation baselines compared against Jev scoring.

— Names baselines but no usage detail v0.4.0

└──▷ BREAKING ON UPGRADE

! The default LLM changed from GPT-4o-mini to GPT-6 Luna; set `VGRAG_LLM_MODEL` explicitly to retain your previous model.

! Custom OpenAI-compatible endpoints must support the Responses interface when using GPT-6 models.

! The OpenAI SDK minimum version is now 1.66.0.

! Existing cache entries are not invalidated, but GPT-6 Responses requests use separate cache keys, so prior cached results will not be reused for GPT-6 calls.

WAS THIS USEFUL? 👍 👎

open-genai

Feedback ⓘ v0.13.0 NOTES ↗ RANK

デジタル庁のガバメントAI「源内(GENAI)」を完全ローカル(ローカルLLM/OpenAI互換)で動かす非公式プロジェクト。SAML認証(Keycloak)・RAG(Qdrant)・文字起こし(Whisper)・画像生成(SD)・チーム単位ナレッジをローカル完結。

> open-genai v0.13.0 adds a self-hosted LMS module, a wizard-driven procedure creation flow, an event scheduling feature, Markdown rendering for chat answers, and per-account API rate limiting.

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

01 **New LMS module for video and PDF courses** NEW 85

Launches an opt-in LMS module started with `docker compose --profile lms up -d`, serving video and PDF courses at the `/lms` route; it is excluded from the default startup profile.

Spin up the LMS module alongside the main stack when you want to host video and PDF courses — it does not start by default.

```
$ docker compose --profile lms up -d
```

— Exact command and route given, mechanism clear v0.13.0

THINNER COVERAGE BELOW

02 **Per-account API rate limiting** NEW 50

Adds per-account API rate limiting for authenticated calls, defaulting to 300 requests per minute.

— Default number given but no config key or flag named v0.13.0

03 **Scheduling feature for events** NEW 50

Adds a scheduling feature for creating events in a structured basic-info → organiser → dates flow, covering both the organiser and respondent workflows.

— Flow stages named but no navigation or command v0.13.0

04 **Wizard-driven procedure creation flow** NEW 45

Adds a wizard that freezes submitted documents and review items at submission time, then continues to a ledger or an existing system after confirmation.

— Describes behaviour but no UI path or config surface v0.13.0

05 **Markdown rendering in chat answers** IMPROVED 35

Renders chat answers as Markdown, with diagrams and images revealed only on tap/click rather than shown inline immediately.

— Describes behaviour change with no further mechanism detail v0.13.0

WAS THIS USEFUL? 👍 👎

garden-skills

Feedback ⓘ motion-showreel-v0.1.0 NOTES ↗

RANK

garden-skills is an open-source collection of agent skills for web design, knowledge retrieval, and image generation.

> garden-skills ships a new motion-showreel agent skill that renders code-driven short videos deterministically, with a 43-style library and synchronized audio/narration support.

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

01 motion-showreel skill for code-rendered videos NEW

93

Introduces the `motion-showreel` agent skill for producing code-rendered short videos (showreels, UI morphs, kinetic typography, logo/title sequences, beat-synced generative loops) output as MP4 plus editable source. Rendering is deterministic via a render(t) pipeline where stills, contact sheets, three-direction boards, proxies, and masters all derive from the same page, with motion blur, supersampling, and film grain exposed as render flags. Ships a 43-style library (including swiss-editorial, ink-wash, risograph, synthwave, soft-3d-toy, cinematic-title, pixel-game, comic-pop, flat-vector, guochao, painterly, glitch-digital, soft-organic) each defining look, motion, camera grammar, and sound palette. Picture and sound share one timeline, with music from a built-in zero-dependency synth or Tone.js with a sampled piano, or cut to a user-supplied track using beat and drop detection; narration comes from Gemini/MiniMax TTS on user scripts or real timestamps from user recordings, with music and SFX yielding to speech. Supports films up to 20 minutes via persistent chapter caches, and enforces technical acceptance checks (size, fps, duration, frame count, black frames, loudness, true peak, loop seam, determinism) separately from creative review. Install with `npx skills add ConardLi/garden-skills/tree/motion-showreel-v0.1.0/skills/motion-showreel`, or download the pinned `.zip` from the release and unpack into `.claude/skills/`, `.agents/skills/`, or `.codex/skills/`.

Install the motion-showreel skill into a Claude Code agent project so it can produce code-rendered MP4 videos.

```
$ npx skills add ConardLi/garden-skills/tree/motion-showreel-v0.1.0/skills/motion-showreel
```

Install via the pinned zip when the skills CLI is unavailable, placing the skill where Claude Code, Codex, or other agents will discover it.

```
$ curl -fsSL -o motion-showreel.zip \
  https://github.com/ConardLi/garden-skills/releases/download/motion-showreel-v0.1.0/motion-showreel-0.1.0.zip
unzip motion-showreel.zip -d .claude/skills/
```

— Names pipeline, style library, audio stack, and exact install commands.

motion-showreel-v0.1.0

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ snapshot-20261025 NOTES ↗

RANK

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

> LangSmith's latest changelog centers on MCP server authentication fixes, new API capabilities for filtering threads and relinking deployment repos, an organization-wide AI search feature, and several breaking changes to deployment UI, Teams API access, and organization permission checks.

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

01 **Deployment repo relinking via API** NEW 80

`PATCH /v2/deployments/<deployment-id>` now accepts a `source_config.repo_url` field, letting an existing deployment be pointed at a renamed or transferred GitHub repository without recreating the deployment.

Point an existing deployment to a renamed or transferred GitHub repository without recreating the deployment.

```
$ curl -X PATCH
'https://api.smith.langchain.com/v2/deployments/<deployment-id>'
\
  -H 'X-API-Key: <your-api-key>' \
  -H 'Content-Type: application/json' \
  -d '{"source_config": {"repo_url": "https://github.com/my-
org/new-repo-name"}}'
```

— Runnable curl example with named endpoint and field snapshot-20261025

02 **Deployment UI tab, header, and deletion changes** BREAKING 70

The separate Revisions tab in Deployment is removed, so bookmarks to tabs after Overview now open the tab to their right. Custom headers for a deployment are no longer shown in the Settings dialog — they now live only in the Connect dialog. Deleting a deployment now requires typing permanently delete to confirm.

— Three UI changes merged with exact before/after behavior snapshot-20261025

03 **MCP server authentication behind tunnels** IMPROVED 65

Custom MCP servers behind a tunnel now accept `x-api-key` and `Cookie` headers the same way servers outside a tunnel do, while authorization endpoints behind a tunnel now return `400`. The hosted LangSmith MCP server also now accepts access tokens issued for the LangSmith API, enabling a single OAuth grant to cover both the API and MCP.

— Names headers and status code but gives no runnable example `snapshot-20261025`

04 Organization-wide AI search for runs and traces NEW

60

Organizations can enable AI search so team members can query runs and traces in plain language, by navigating to Organization Settings > Projects, clicking `Turn on AI search`, reviewing a data-processing and PHI consent step, and clicking Enable.

Enable AI search for your organization so team members can query runs and traces in plain language.

> 📍 In Organization Settings, navigate to Projects, click `Turn on AI search`, review the data-processing and PHI consent step, and click Enable.

— Clear UI navigation steps but no API or config surface `snapshot-20261025`

THINNER COVERAGE BELOW

05 Teams API access gated by setting BREAKING

45

The Teams APIs now return `403` unless the organization has the `can_use_teams` setting enabled.

— Named setting and status code but no endpoint path `snapshot-20261025`

└──▷ BREAKING ON UPGRADE

- ! The separate Revisions tab in Deployment is removed; bookmarks to tabs after Overview now open the tab to their right.
- ! Custom headers for a deployment are no longer shown in the Settings dialog — they now live only in the Connect dialog.
- ! Deleting a deployment now requires typing `permanently delete` to confirm.
- ! The Teams APIs return `403` unless the organization has the `can_use_teams` setting enabled.
- ! Organization actions now require dedicated per-action permissions instead of the single broad `manage` permission; `PATCH /orgs/current/info` checks the permission for each field individually, so callers that previously relied on the broad permission may receive permission errors.

WAS THIS USEFUL? 👍 👎

UK AI Security Institute Inspect AI Feedback ⓘ

0.3.284 CODE ↗

Inspect is the UK AI Security Institute's framework for large language model evaluations, covering datasets, solvers, scorers, and agentic eval tasks.

Adds sentinel `step.agent_messages`, asciicast playback under `asciinema:*` sample-store keys, and server-side S3 log copy on eval retry.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch 0.3.284
https://github.com/UKGovernmentBEIS/inspect_ai.git
# already have the repo? check out this version:
$ git checkout 0.3.284
```

Adds `step.agent_messages` to the sentinel API, exposing the running agent's own message list at every stage of execution.

Viewer plays asciicasts stored in the sample store under `asciinema:*` keys.

Retrying an eval whose logs are on S3 now copies the prior log server-side and uploads only the new tail, cutting retry start time from ~16–29 s to ~2 s for large logs.

NEW INTEGRATION Kimi K3.1 Pro and K3.1 Pro Highspeed via moonshot provider

UPDATED INTEGRATION Fireworks and Together model info updated

WAS THIS USEFUL? 👍 👎

overmind

Feedback ⓘ v0.1.15 NOTES ↗ RANK

The platform for continuously improving AI agents.

> Overmind's four releases this window centered on decision model training and serving, an agentless Data Workshop, and TypeScript repo tracing, alongside smaller CLI, SDK and Console refinements.

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

01 MCP config handling in init and sync IMPROVED

75

`overmind init` now writes a URL-only MCP config file, and `overmind sync` leaves existing MCP and `.git` entries untouched when syncing.

— Exact commands named, behavior clearly stated v0.1.15

02 TypeScript repo scanning with OTLP trace export NEW

75

Overmind now scans TypeScript repositories and exports their JS traces over OTLP, letting TypeScript agent instrumentation report traces the same way as other languages.

Instrument a TypeScript agent repo so production traces are exported over OTLP and appear in Overmind.

```
$ pip install "overmind[tracing]"
export OVERMIND_API_KEY=ovr_…
overmind sync
```

— Includes a runnable install and sync command v0.1.7

03 Agentless Data Workshop and dataset versioning IMPROVED

75

Data Workshop can now run without an agent, using MCP versions, sandboxed runs, and document support. Datasets can be created from versions with script attachments, version analytics, and cancellation support, and dataset imports are now batched with reduced Workshop validation I/O.

— Several named mechanisms grouped but no exact commands v0.1.7 v0.1.4

04 Decision model training, serving and workflow integration NEW

70

Decision models can now be trained and served directly from Workshop datasets, including serving on the TypeSafe `systemone` endpoint. Native workflows and

`DecisionJob` identity are now integrated on the main infrastructure, and trained model decisions are surfaced inside inference code examples.

— Names endpoint and identity concept but no setup steps v0.1.7 v0.1.4

05 **Optional project argument in create_api_key** IMPROVED

60

The `project` argument in `create_api_key` is now optional, allowing API keys to be created without specifying a project.

— Names the exact function and argument changed v0.1.12

THINNER COVERAGE BELOW

06 **Parallel, resumable OCR with automatic language selection** NEW

45

OCR now runs in parallel and is resumable, with automatic language selection.

— Mechanism named but no usage path given v0.1.7

07 **Console onboarding, theme and loading UI refresh** IMPROVED

40

Console onboarding has been unified into a single flow, the Console now runs in dark-only mode, and content loading states use skeleton loaders.

— UI changes named but no navigation path given v0.1.7 v0.1.4

08 **Bounded signal scan memory usage** IMPROVED 30

Signal scan memory usage is now bounded without needing to split conversations.

— Thin one-line description with no mechanism detail v0.1.7

09 **Optional SDK analytics with stricter capability scans** IMPROVED

30

SDK analytics are now optional, and the SDK requires more detailed capability scans.

— Brief statement with no config key or flag named v0.1.4

WAS THIS USEFUL? 👍 👎

Langfuse

Feedback ⓘ v4.70.0 NOTES ↗ RANK

Langfuse provides tracing, evaluation, and monitoring for LLM applications.

> Langfuse rolled out a self-hosted AI Gateway with guided setup and per-member key management, expanded its new Topics analysis feature with a public API and cluster/treemap views, and added several MCP server tools alongside smaller tracing, export and UI improvements.

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

01 **AI Gateway deployment configuration and key management** NEW

95

Adds `LANGFUSE_AI_GATEWAY_DEPLOYED` and `LANGFUSE_AI_GATEWAY_PUBLIC_URL` environment variables to gate and configure the self-hosted AI gateway, showing a deploy placeholder when absent and allowing the displayed gateway URL to be overridden, plus a guided AI gateway setup flow with per-tool setup instructions shown after creating a gateway key; also adds a global enable flag and derived JWT public key / key ID for the AI gateway when self-hosting, allows project members to create their own AI gateway API keys while owners and admins can issue gateway keys as a service account, and adds an opt-in plain HTTP web URL for in-cluster AI gateway traffic.

Enable the self-hosted AI gateway and point the UI at your public gateway URL so team members see the correct endpoint in setup instructions.

```text
LANGFUSE_AI_GATEWAY_DEPLOYED=true
LANGFUSE_AI_GATEWAY_PUBLIC_URL=https://gateway.example.com
```

— Named env vars with concrete config example Describes mechanism and roles but no exact flag names given

v4.69.0 v4.68.0

02 **Topics analysis feature: API and UI views** NEW

75

Adds unstable read-only Topics endpoints to the public API (`GET /api/public/topics`) for programmatic access to topic data, Treemap, Grid, and cluster-map views for Topics analysis results in the web UI, and a guided first-run experience for Topics with combined backfill and progress tracking.

Fetch topic data programmatically using the new read-only Topics public API endpoint.

```
$ curl -X GET 'https://<your-langfuse-host>/api/public/topics' \
  -H 'Authorization: Bearer <your-secret-key>'
```

— Named endpoint with curl example; UI views named but undetailed v4.70.0 v4.68.0

03 **MCP server tool and management additions** NEW 65

Adds `runBatchEvaluation` MCP tool and exposes `batch_action_id` on batch run outputs in the evals subsystem, adds skill version creation and label/tag management in the MCP interface, and adds a `listLlmConnections` tool letting agents enumerate configured LLM connections.

— Named MCP tools and fields but no usage example given v4.70.0 v4.69.0 v4.68.0

THINNER COVERAGE BELOW

04 **Tracing view skill filters and topic hover cards** IMPROVED 45

Enhances the tracing UI with available and loaded skill name filters, requested skill loads and file paths shown in the tracing view, and a trace's topics shown in a trace header hover card.

— Describes UI additions without exact navigation path v4.69.0

05 **Cmd+K quick navigation to more resources** IMPROVED 45

Extends Cmd+K quick-navigation to jump directly to prompts, datasets, queues, and evaluators.

— Clear shortcut named but no further mechanism v4.70.0

06 **Outbound host allowlist for blob storage** NEW 45

Adds an outbound host allowlist for blob storage endpoints to restrict which external hosts the service may reach.

— Mechanism stated, no config key or flag named v4.69.0

07 **Score metadata and provenance fields in blob export** NEW 45

Adds opt-in `score metadata` and `provenance` field groups to blob export.

— Named field groups but no config flag or example v4.68.0

08 **Legacy ingestion API restricted for new organizations** BREAKING 40

Limits legacy ingestion API to scores only for new organizations.

— States scope and effect but no endpoint path named v4.68.0

09 **Syntax highlighting for setup snippets** IMPROVED 30

Adds syntax highlighting for shell, TOML, and JSON setup snippets in the web UI.

— Minor UI polish, no further detail v4.70.0

10 **Managed skill reuse loaded from Langfuse** NEW 30

Loads managed skills from Langfuse in the agent, enabling centrally-defined skill reuse.

— Describes capability without mechanism or config detail v4.68.0

11 **Independent control of events table UI sections** IMPROVED 30

Allows controlling events table UI sections independently.

— Bare description, no exact UI control named v4.68.0

12 **Versioned input storage for trace batching** NEW 25

Adds versioned input storage for trace batching.

— Bare one-line addition, no mechanism detail v4.68.0

13 **Execution issue collection in agent transcripts** NEW 20

Collects execution issues alongside the assembled transcript for easier debugging.

— Bare description, no mechanism or surface named v4.69.0

WAS THIS USEFUL? 👍 👎

Arize Phoenix

Feedback ⓘ arize-phoenix-v20.26.0 NOTES ↗

RANK

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

> Phoenix v20.26.0 adds richer OpenTelemetry attribution and streamlines its factory triage workflow with Claude Console linking and automatic dispatch of trusted candidates.

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

01 **OpenTelemetry service.version resource attribute** IMPROVED

55

The Phoenix server's OpenTelemetry resource now includes `service.version`, giving richer telemetry attribution for traces emitted by the server.

— Names the exact OTel attribute but gives no further mechanism. arize-phoenix-v20.26.0

02 **Claude Console links in factory comments** NEW

35

Session IDs referenced in factory comments are now linked directly to the Claude Console, allowing faster navigation during triage.

— Behavior described but no UI path or config detail. arize-phoenix-v20.26.0

03 **Automatic dispatch of trusted factory candidates** NEW

35

Phoenix now nominates factory candidates during triage and automatically dispatches the ones deemed trusted.

— No criteria or mechanism for trust is specified. arize-phoenix-v20.26.0

WAS THIS USEFUL? 👍 👎

eunomia-bpf AgentSight

Feedback ⓘ v1.0.35 NOTES ↗

RANK

AgentSight is an eBPF observability tool that profiles AI agent activity by correlating LLM traffic, process execution, file access, and network events without SDK integration.

> AgentSight v1.0.35 adds PID-namespace-scoped sidecar capture and surfaces TLS transport identity at the eBPF level, improving attribution of encrypted traffic without SDK integration.

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

01 **eBPF-level TLS transport identity exposure** NEW 45

Exposes TLS transport identity and capture metadata via eBPF, enabling attribution of encrypted traffic to specific processes and connections.

— Explains mechanism but no concrete API or config surface named. v1.0.35

02 **PID-namespace-scoped sidecar capture** NEW 25

Adds PID-namespace-scoped sidecar capture, as named in the release summary; no further mechanism or configuration detail is given.

— Named only in summary, no detail on behavior or usage. v1.0.35

WAS THIS USEFUL? 👍 👎

Comet Opik

Feedback ⓘ 2.2.118 NOTES ↗ RANK

Opik traces, evaluates, and monitors LLM applications and agentic workflows, with datasets, an LLM-as-judge metric library, and production dashboards.

> Opik made dashboard editing and Ollie widgets easier to discover in its dashboard UI.

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

01 **Ollie widgets discoverable in dashboard editor** IMPROVED

37

Dashboard editing and Ollie widgets are now surfaced directly from the Add widget menu in the Opik dashboard UI.

– Names exact UI menu but no further mechanism detail 2.2.118

WAS THIS USEFUL? 👍 👎

Weights & Biases Weave

Feedback ⓘ v0.53.13 NOTES ↗ RANK

Weights & Biases Weave traces, evaluates, and monitors LLM applications for developers.

> Weave v0.53.13 updates its OpenTelemetry instrumentation to emit the current OTel GenAI semantic convention attribute names for LLM call spans.

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

01 **Current OTel GenAI attribute names for LLM spans** IMPROVED

35

Weave now emits the current OpenTelemetry GenAI semantic convention attribute names for LLM call spans, replacing older attribute naming.

– Names the OTel convention but no exact attribute names or migration steps given

v0.53.13

WAS THIS USEFUL? 👍 👎

suvadu

Feedback ⓘ v0.5.2 NOTES ↗ RANK

Shell history replacement with AI agent tracking and MCP server. Built in Rust, 100% local.

> Suvadu shipped a searchable `suv home` feature map and settings for it, cargo binstall distribution with minisign verification and an MCP Registry listing, and a broad set of capture/recall improvements including failure-output capture, Antigravity agent recording, git-branch tracking, bookmark placeholders, and a sudoless install script.

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

01 **Failure output capture for failed commands** NEW 90

Setting `[capture] failure_output = true` in `config.toml` retains the last 4 KB of a failed command's output, which is then surfaced in recall and MCP tools.

Capture why a command failed so recall and MCP tools can show the last 4 KB of its output.

toml

```
[capture]
failure_output = true
```

— Exact config key, size limit, and surfaced locations given v0.5.1

02 **Sudoless install script with flags and signature verification**

IMPROVED 90

The install script adds `--user` and `--dir DIR` flags for installing without sudo, with `SUVADU_INSTALL_DIR` as an environment-variable alternative. It now adds the shell hook automatically (with confirmation and an rc-file backup) and verifies the release signature with `minisign`.

Install Suvadu into ~/.local/bin without requiring sudo, in a shared or restricted environment.

```
$ curl -fsSL https://downloads.appachi.tech/suvadu/install.sh | bash -s -- --user
```

— All flags, env var, and verification mechanism named with command v0.5.0

03 **Cargo binstall support and MCP registry listing** NEW 85

Suvadu now supports `cargo binstall suvadu`, which downloads a signed pre-built binary and verifies its minisign signature with the same key `suv update` uses. The `suvadu` command alias was added alongside `suv` when installed via `cargo install suvadu`, so `suvadu mcp-serve` works regardless of install method, and Suvadu was registered in the official MCP Registry as `io.github.AppachiTech/suvadu`.

Install Suvadu without compiling from source, getting a minisign-verified pre-built binary in seconds.

```
$ cargo binstall suvadu
```

— Names exact commands, key use, and registry id with runnable example v0.5.2

---

04 **Antigravity agent command recording via zsh hook** NEW

85

`suv init antigravity` installs a `~/.zshenv` capture script that records commands run by Antigravity's agent inside `zsh -c` subshells. The command adds a marked block to `~/.zshenv`, backs up the previous file, and requires restarting Antigravity to take effect.

Record Antigravity agent commands by installing its zsh capture hook, then restart Antigravity.

```
$ suv init antigravity
```

— Command, file touched, and restart requirement all named v0.5.1

---

05 **suv home feature map and startup screen setting** NEW

85

`suv home` (or a bare `suv`) opens a searchable map of all Suvadu features, grouped by task, showing the real command behind each. A new Startup Screen option in `suv settings` (set to `help`) restores the old command overview instead of this new Home screen on a bare `suv`.

Explore everything Suvadu can do interactively — useful when onboarding a new team member or discovering less-used features.

```
$ suv home
```

— Runnable command plus named settings path to revert v0.5.0

---

06 **Daily background update check with opt-out** NEW

85

Suvadu now performs a once-a-day background release check; it can be disabled via `[update] check = false` in `config.toml` or the `SUVADU_NO_UPDATE_CHECK=1` environment variable.

Opt out of the daily background release check in environments where outbound requests must be suppressed.

```toml
[update]
check = false
```

— Both opt-out mechanisms named with exact config/env values v0.5.0

07 **Recall highlighting, grouped view columns, and defaults**

IMPROVED 85

Recall now highlights matched characters in every result row to show why each command matched. The grouped recall view (Ctrl+R / `suv search`) shows Last used and Runs columns for each grouped command, and a new Recall defaults section in `suv settings` sets the default match mode, starting scope, and compact recall.

— Names keybinding, columns, and settings section precisely v0.5.0

08 **Zsh-autosuggestions strategy backed by Suvadu** NEW 80

Setting ZSH_AUTOSUGGEST_STRATEGY=(suvadu history) wires Suvadu's ranked recall into zsh-autosuggestions, falling back to shell history when Suvadu has no suggestion.

Get Suvadu-ranked suggestions in zsh-autosuggestions, falling back to shell history when Suvadu has nothing.

```zsh
ZSH_AUTOSUGGEST_STRATEGY=(suvadu history)
```

— Exact config line and fallback behaviour given v0.5.1

09 **Bookmark placeholders with prompt-at-pick values** BREAKING 80

Bookmarks now support placeholders such as `ssh {{user:root}}@{{host}}`, prompting for values when the bookmark is picked; escape with `\{{word}}` to keep a literal placeholder. This changes prior behaviour: a bookmark containing a literal `{{word}}` will now prompt for a value on selection unless escaped.

— Syntax, escape rule, and behaviour change all specified v0.5.1

10 **(new) Recall detail pane: scrolling and git branch tracking**

NEW

75

Recall's detail pane now supports `Shift+Tab` to scroll its contents and shows the git branch each command ran on, which can be disabled with `[shell] capture_git_branch = false` in `config.toml`.

– Bare keybinding addition with no further context Config key and UI location both named v0.5.1

---

11 **Project-scoped skills via .suvadu/skills** NEW

70

Adds support for `.suvadu/skills/*.md` project-scoped skills, which must be trusted explicitly with `suv skills trust`.

– Names file path and trust command but little mechanism v0.5.1

---

12 **Git snapshot capture for agent session handoff** NEW

65

Agent sessions (Claude Code, Codex, OpenCode, Cursor) now capture a git snapshot of changed file paths at each turn, used for `what_changed` and handoff reporting.

– Names integrated agents and output use but no direct command v0.5.1

THINNER COVERAGE BELOW

13 **list_snippets MCP tool for bookmarked snippets** NEW

50

Adds the `list_snippets` MCP tool so agents can discover bookmarked command templates.

– Names the tool but no usage example or mechanism detail v0.5.1

---

14 **suv doctor Up-arrow binding diagnostic** IMPROVED

50

`suv doctor` now reports when Up-arrow is no longer bound to Suvadu's widget, including the cause and fix.

– Names the check but not the actual fixes it reports v0.5.1

---

15 **Rust 1.95 minimum for cargo install** BREAKING

40

`cargo install suvadu` now requires Rust 1.95 or newer.

– Clear version bump but no migration guidance v0.5.1

└──▷ BREAKING ON UPGRADE

! A bookmark containing a literal `{{word}}` now prompts for a value on selection; write it as `\{{word}}` to preserve the literal braces.

! `cargo install suvadu` now requires Rust 1.95 or newer.

! `suv init antigravity` adds a marked block to `~/.zshenv`, backing up the previous file; a restart of Antigravity is required.

WAS THIS USEFUL? 👍 👎

agentic-os

Feedback ⓘ v1.8.35 NOTES ↗ RANK

Governance framework for AI coding agents with a five-step workflow (plan, build, review, test, ship) requiring evidence at each stage. Drop-in rules via AGENTS.md.

> agentic-os tightened its evidence-based ship pipeline with a test-run recorder plus a freshness validator, a stricter AC-6 Test Gate Results check, and hardened capabilities-gate and KB-consultation path resolution, alongside deploy-script fixes for `CP_FLAG` tracking, `--ref` pinning, and unknown-flag handling.

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

01 **Test evidence freshness validation in ship pipeline** NEW 93

`.agentcortex/tools/record_test_run.py run --worklog <log> -- <test command>` runs a test suite without invoking a shell and records the exit code plus a code fingerprint. A new test evidence freshness validator line fails a current-branch Work Log whose latest recorded run failed before a PASS receipt, or whose code changed after a passing run before shipping, and warns when a tested-tier log has a PASS receipt but no recorded run — surfaced by both `validate.sh` and `/ship`.

— Full command, validator logic and integration points given. v1.8.35

02 **Pin deploy updates to a release tag** NEW 70

`deploy_brain.sh --ref <tag>` (or `-Ref` in PowerShell, or the `ACX_REF` environment variable) pins an update to a specific release tag instead of always pulling the latest.

— Exact flag and env var make it directly usable. v1.8.35

03 **CP_FLAG deploy skip/decline tracking** IMPROVED 70

When a copy is declined or skipped via `CP_FLAG=-n` or answering `no` to `CP_FLAG=-i`, deploy now prints `[KEPT]` status output and a `N kept by CP_FLAG` summary count, while preserving the previous manifest hash so the next deploy still detects the difference.

— Named flag values and manifest hash behavior explained. v1.8.34

04 **KB-routed task consultation** IMPROVED 68

When `downstream-capabilities.yaml` declares `knowledge_sources`, `AGENTS.md` and `CLAUDE.md` now instruct agents to consult the KB before planning or reviewing a change in a KB-routed domain, without requiring `/bootstrap`. `kb-consult` preserves the KB's own routed order instead of resorting entries by `approx_tokens`, which still acts as a consult size limit.

— Named config key and files but no exact invocation. v1.8.34 v1.8.35

05 **deploy.sh `--help` and unknown-flag rejection** BREAKING

65

`deploy.sh --help` now prints usage. An unknown flag now exits with code 2 instead of silently creating a directory named after the flag — any script that previously passed an unsupported flag deployed into a bogus directory and will now fail outright. `deploy.ps1 --help` likewise no longer deploys into the current directory.

— Breaking behavior and flag named, migration implied. v1.8.35

06 **Configurable capabilities gate path validation** BREAKING

65

The capabilities gate-safety validators now read `downstream_capabilities.path` from config instead of hardcoding `.agentcortex/context/private/downstream-capabilities.yaml`, so a retargeted path is actually checked. If that path is configured, a flow mapping, tab indent, or repeated key in that section now fails closed rather than passing silently.

— Named config key and default path, behavior change clear. v1.8.34

07 **Stricter AC-6 Test Gate Results check** BREAKING

60

The AC-6 Test Gate Results check now requires a non-empty body that is not the literal `none`, rather than merely the presence of the heading. A Work Log at handoff/ship whose Test Gate Results body is empty or reads `none` now produces a FAIL where it previously passed.

— Clear before/after but no remediation command given. v1.8.34

THINNER COVERAGE BELOW

08 **Model inheritance for acx-* subagents** NEW

55

`acx-*` subagents can now set `model: inherit` to use the session's model instead of being pinned to opus or sonnet, so a haiku session is no longer silently upgraded.

— Named config value but no file path given. v1.8.35

09 **Guardrail and retro workflow policy updates** IMPROVED

45

`engineering_guardrails` §8 now specifies that a standing instruction (e.g. `keep going`, blanket permission to merge) does not satisfy an owner decision such as freezing a spec or amending an ADR. `security_guardrails` §2 now requires reviewing LLM output as untrusted input. The `/retro` workflow now retires a lesson first when its root cause is already enforced by a guardrail.

— Policy text changes named but no concrete surface. v1.8.35

10 **Governance gated behind .acx-incoming sidecar merge** BREAKING

45

When a project already has its own `AGENTS.md`, `CLAUDE.md`, or `GEMINI.md`, deploy and both validators now report that governance is not active until the `.acx-incoming` sidecar is merged.

— Named files and sidecar but no merge procedure given. v1.8.35

└──▷ BREAKING ON UPGRADE

- ! `deploy.sh` now rejects unknown flags with exit 2 and discards them; any script that passed an unsupported flag previously deployed into a directory named after that flag — those invocations will now fail.
- ! `deploy.ps1 --help` no longer deploys into the current directory.
- ! When a project has its own `AGENTS.md`, `CLAUDE.md`, or `GEMINI.md`, deploy and both validators now report that governance is not active until the `.acx-incoming` sidecar is merged.
- ! If `downstream_capabilities.path` is configured, the validators now check that file; a flow mapping, tab indent, or repeated key in that section now fails closed rather than passing silently.
- ! A Work Log at handoff/ship whose Test Gate Results body is empty or reads `none` now produces a FAIL where it previously passed.

WAS THIS USEFUL? 👍 👎

openlit

Feedback ⓘ otel-gpu-collector-1.0.0 NOTES ↗ RANK

OpenLIT is the open-source agent harness engineering platform: trace, evaluate, guard, and improve everything around the model in your AI agents, on OTEL.

> OpenLIT's OpenTelemetry GPU Collector 1.0.0 refines disk metric collection on Linux to avoid errors from read-only block devices.

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

01 **Read-only disk handling in GPU collector** IMPROVED 45

The OpenTelemetry GPU Collector skips write disk metrics for read-only Linux block devices, resuming reporting when a device becomes writable again. Missing or invalid read-only indicators are treated as writable so metrics continue to be collected by default.

– Explains mechanism and fallback behaviour but no config surface named

otel-gpu-collector-1.0.0

WAS THIS USEFUL? 👍 👎

heron

Feedback ⓘ v0.8.3 NOTES ↗ RANK

Agent and LLM API performance monitoring via network packet probe. Measures performance of OpenClaw, Claude, Codex, DeepAgents and more — deployed on the provider side, no SDK changes required.

> Heron v0.8.3 reworks its Performance page to show prefill and decode token throughput rates, overhauls internal active-call tracking metrics, and ships a breaking config fix for aglake-props deployments.

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

01 props.toml models_used field rename for aglake-props

BREAKING

80

Deployments that previously ran `heron aglake-props` and merged its output into `props.toml` must change the `models_used` line to `models_used{}` and restart aglaked; the corrected field name applies only to newly written buckets.

— Exact command, config file, field rename, and restart step given. v0.8.3

02 Prefill/decode token rate charts on Performance page

IMPROVED

65

The Performance page's Cache Token Usage and Token Averages charts are replaced with `Prefill (Input) Tokens / s` and `Decode (Output) Tokens / s` rate charts. Cache reads and writes are now drawn beside input rather than stacked on top of it, so throughput reads consistently at any zoom level.

— Names exact charts replaced and zoom behaviour; only a UI path, no command. v0.8.3

03 Active call tracking via connection identity

BREAKING

60

New internal metrics `active_calls_open` and `active_calls_expired` replace `aggregator_concurrency_table`, exposing in-flight call counts tracked by connection identity rather than by dimension.

— Names metrics and mechanism shift but no usage surface for readers. v0.8.3

└──▷ **BREAKING ON UPGRADE**

! Deployments that previously ran `heron aglake-props` and merged its output into `props.toml` must change the `models_used` line to `models_used{}` and restart aglaked; the corrected field name applies only to newly written buckets.

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ snapshot-20261025 NOTES ↗ RANK

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

> Braintrust's October 2026 release adds Cursor agent tracing, resumable Go workflow evaluations, TypeScript application logs, continuous Loop automations, a coding agent insights dashboard, MCP-based human review and dataset copying, and a new conversation-turn thread view.

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

01 **Coding agent insights dashboard** NEW 86

A new public-preview dashboard at `Dashboards > Coding agent insights` shows coding agent usage by user, API key, model, tool, and branch with cost breakdowns, plus a daily Loop automation that surfaces recurring problems; requires data plane v2.13.0+.

Reach the Coding agent insights dashboard to review per-user and per-model cost breakdowns for your team's coding agents.

📍 1. In the left navigation, go to Dashboards › Coding agent insights. 2. Review the usage breakdown by user, API key, model, tool, and branch. 3. Expand a row to see cost breakdowns by user or model.

— Exact navigation path and breakdown dimensions given. snapshot-20261025

02 **Cursor agent session tracing via CLI** NEW 83

`bt trace enable cursor` (requires `bt` v0.26.0+) traces Cursor agent sessions from the CLI and desktop app, capturing turns, model calls, tool calls, and subagents.

Set up full Cursor agent session tracing — turns, model calls, tool calls, and subagents — so you can inspect them in Braintrust alongside other traces.

```
$ bt trace enable cursor
```

— Exact command, version requirement, and capture scope given. snapshot-20261025

03 **TypeScript application logs with severity methods** NEW 81

TypeScript SDK v3.39.0+ adds application logs via severity methods such as info() and error() on a logger, or wrapConsole() to forward existing console output alongside traces.

Forward your existing Node.js console output into Braintrust application logs so log lines appear alongside traces without changing every call site.

```
$ import { wrapConsole } from 'braintrust';
wrapConsole();
console.info('Starting evaluation run');
```

— Named methods plus a working code snippet. snapshot-20261025

04 **MCP server support for human review and dataset copying** NEW

69

The Braintrust MCP server can send project log traces, experiment traces, or dataset rows to human review with optional reviewer assignment (requires data plane v2.19.0+); MCP server and Loop can also copy project log rows into datasets while preserving input, metadata, tags, and source log links (requires data plane v2.18.0+).

— Two capabilities with version gates but no example or exact command. snapshot-20261025

05 **Go workflow evaluations with resumable runs** NEW 67

Go SDK v0.18.0+ adds workflow evaluations that let you start a run and resume it later by polling or from a webhook, including a Redis store for cross-process resume, with results recorded in a single experiment.

— Mechanism and SDK version named but no runnable example given. snapshot-20261025

THINNER COVERAGE BELOW

06 **Continuous mode for scheduled Loop automations** NEW

50

Continuous mode lets scheduled Loop automations check for new traces between runs and analyze them in the current thread once enough arrive; requires data plane v2.20.0+.

— Behavior described but no setup steps or UI path. snapshot-20261025

07 **Thread view for agent traces** NEW 45

A new public-preview thread view organizes agent traces by conversation turn, showing the request, agent work, and final response per turn.

— Describes what it shows but no navigation or config detail. snapshot-20261025

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI Runtime Defense & Guardrails

FailproofAI

Feedback ⓘ fp-cloud-cli-v0.0.2 NOTES ↗ RANK

Observability and enforcement for AI agent harnesses. Capture every run and runtime reliability with policy enforcement.

> FailproofAI moved Jev policy evaluation to the cloud this window, adding full author/publish/backtest tooling and per-agent fleet deployment scoping, alongside browser-based Cloud login with automatic skill installation and new CLI surfaces for issues, session import, and permissions.

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

01 Fleet deployment scoping, history, and rollback NEW

95

Adds `fp fleet deploy --target POLICY=INTEGRATION[/agt_ID]` (repeatable for multiple alternatives) and `--all-agents POLICY` to scope a Cloud policy to a single integration, a specific agent profile, or clear scope across a whole machine. `fp fleet history` and `fp fleet rollback` now include each deployment's Jev mode, and a new `fp fleet show` surface lets a machine report Cloud policies it could not apply — failed loads, unsupported Jev modes, or checks dropped for the question budget.

Scope a fleet deployment so only a specific agent instance receives the policy, then confirm the assignment.

```
$ fp fleet deploy --target my-regex-policy=claude/agt_abc123 --
target my-jev-policy=claude
```

Publish a hybrid Cloud policy that lets Jev checks review your JavaScript rules, then deploy it to a single Cursor agent profile on one integration.

```
$ fp policies publish --kind both
fp fleet deploy --target my-policy=cursor/work-profile
```

Deploy a Jev-only policy across all agent profiles on a machine when you want cloud-side Jev enforcement without touching local config.

```
$ fp fleet deploy --all-agents my-jev-policy
```

Review deployment history including each rollout's Jev mode, then roll back a problematic deployment.

```
$ fp fleet history
fp fleet rollback
```

— Multiple named subcommands and flags with runnable examples. `v1.0.12`

`fp-cloud-cli-v0.0.2`

---

02 **Jev policy backtesting against historical traffic** NEW

90

Adds `fp policies backtest` with `--kind`, `--semantic`, `--hours`, `--agent-id`, `--env`, `--run`, and `--sample N` flags — runs a free cost estimate by default or spends Jev allowance with `--run`; accepts a published `jev` / `both` policy or `--semantic checks.json`.

Estimate how a published Jev policy performs against historical traffic before spending budget — run without `--run` for a free preview.

```
$ fp policies backtest pol_abc123 --kind jev --hours 48 --agent-id agt_xyz --env production
```

— Full flag set and cost-estimate mechanism with runnable example. `fp-cloud-cli-v0.0.2`

---

03 **Cloud policy publishing for regex, Jev, and hybrid checks** NEW

86

Adds `fp policies publish <id> --kind regex|jev|both [--source f.mjs] [--semantic f.json]` to publish Cloud policies of three kinds: pure JavaScript (`regex`), Jev-only (`jev`), or hybrid (`both`) where Jev checks review the JavaScript; the CLI exits 2 before sending if `--kind` contradicts the given inputs.

Publish a hybrid Cloud policy that lets Jev checks review your JavaScript rules, then deploy it to a single Cursor agent profile on one integration.

```
$ fp policies publish --kind both
fp fleet deploy --target my-policy=cursor/work-profile
```

— Names exact flags, kinds, and validation behavior with runnable example. `v1.0.12`

`fp-cloud-cli-v0.0.2`

---

04 **`jev:evaluate` permission and machine key preset** NEW

85

Adds `jev:evaluate` as a known permission accepted by `fp keys` and `fp users`; the `admin` preset carries it, and the server enforces presence of `events:add` and `policies:pull` or returns 422 naming missing permissions. Adds `fp keys create/update --permission-set machine` for the dashboard's key-only preset (`events:add` + `policies:pull` + `jev:evaluate`), with CLI-side validation that exits 2 naming missing prerequisites before sending.

— Exact permission names, error codes, and preset composition given. `fp-cloud-cli-v0.0.2`

---

05 **Browser-based Cloud login and skill management via `failproofai config`**

NEW

80

Extends `failproofai config` with browser-based login — on a connected machine it opens a browser to sign in or up, pick or create an org, and authorize the machine, while over SSH it prints a link to open on any device — and to inject a `failproofai` skill into every supported agent it finds: Claude Code, Codex, Copilot, Cursor, OpenCode, Pi, every Hermes and OpenClaw profile, droid, Devin, Antigravity, and Goose, pointing each at the FailproofAI skills on GitHub. Adds `failproofai uninstall` to remove only the `failproofai` skills that `config` wrote, leaving any manually installed skill intact.

— Names the commands and full agent list but flow is UI-driven, no example. `v1.0.12`

---

06 **Local protocol upgrade requires daemon reinstall**

BREAKING

80

The CLI, daemon, and Hermes plugin now use a new local protocol to carry agent identity; run `failproofai config` once after upgrading to reinstall and restart the daemon — an older daemon's answers are refused rather than silently losing the agent profile. On 1.0.11 a `jev -only` policy silently does nothing, so every machine you deploy Jev or agent-scoped policies to must be upgraded to 1.0.12.

— Exact migration command and failure mode on old versions given. `v1.0.12`

---

07 **Session import for transcripts and event logs** NEW

75

Adds `fp sessions import FILE... --format --env --dry-run`, importing Claude Code transcripts, Codex rollouts, or event JSONL.

Import a batch of Claude Code transcripts into FailproofAI Cloud for retrospective analysis without writing any data.

```
$ fp sessions import ./transcripts/*.jsonl --format claude-code
--env staging --dry-run
```

— Named flags and supported formats with a runnable dry-run example. `fp-cloud-cli-v0.0.2`

---

08 **Issues inbox listing, counting, and filter facets** NEW

72

Adds `fp issues list|count` with inbox filters `--agent-id`, `--env`, `--machine`, `--source`, `--audit-id`, `--assignee`, `--archived`, `--search`, `--since`, and `count --alert-id`, plus `fp issues facets` to list the values inbox filters can take.

— Every filter flag is named but no example command given. `fp-cloud-cli-v0.0.2`

09 **Cloud-side Jev execution with enforce and watch modes** IMPROVED

65

Jev checks now run on FailproofAI Cloud rather than the local machine, so no Jev-related software is installed locally; FailproofAI Cloud sets each machine's Jev mode (enforce when a Jev policy is deployed, off otherwise), and Cloud policies can also be set to watch (observe) mode, which records what a policy would have blocked without actually blocking.

— Explains mechanism and modes but no command or config to act on. `v1.0.12`

THINNER COVERAGE BELOW

10 **AI-drafted Jev policy authoring** NEW

50

Adds `fp policies author --kind regex|jev|both` to request AI-drafted Jev checks or a hybrid policy from the assistant.

— Command named but behavior described only briefly. `fp-cloud-cli-v0.0.2`

11 **Agent profile roster for per-agent policy targeting** NEW

50

The daemon now maintains a roster of up to 256 agent profiles it has seen and reports them to FailproofAI Cloud, enabling per-agent policy targeting.

— States the limit and purpose but no direct interface for the reader. `v1.0.12`

12 **List helper subcommands for trigger events, eval envs, metric keys**

NEW

50

Adds `fp list trigger_events`, `fp list eval_envs`, and `fp list metric_keys` subcommands.

— Three subcommands named but no explanation of their output. `fp-cloud-cli-v0.0.2`

└──▷ **BREAKING ON UPGRADE**

! The CLI, daemon, and Hermes plugin use a new local protocol to carry agent identity; run `failproofai config` once after upgrading to reinstall and restart the daemon — an older daemon's answers are refused rather than silently losing the agent profile.

! On 1.0.11, a `jev -only` policy silently does nothing; every machine you deploy Jev or agent-scoped policies to must be upgraded to 1.0.12.

WAS THIS USEFUL? 👍 👎

◆ AI/LLM Security

cplt

Feedback ⓘ 2026.10.24-160000-c742e86 NOTES ↗ RANK

cplt is a kernel-level sandbox for AI coding agents, running Copilot CLI, Claude Code, OpenCode, Gemini CLI, and others with git and gh guards.

> cplt expanded its sandboxing surface this window with a new MCP host broker for local stdio servers (macOS, then Linux under bubblewrap isolation), GitHub MCP support for OpenCode via its sandbox proxy, a `--deny-env` flag to strip environment variables, and launch-time controls including `--allow-write-state` and quieter profile-change prompting.

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

01 GitHub MCP support for OpenCode via sandbox proxy NEW

80

Adds GitHub MCP support for OpenCode through `cplt`'s proxy, enabling sandboxed MCP access when running OpenCode; run via `cplt --agent opencode --with-proxy`.

Run OpenCode with GitHub MCP access routed through cplt's sandbox proxy to keep credentials protected.

```
$ cplt --agent opencode --with-proxy
```

— Includes a runnable command demonstrating the feature 2026.10.23-053641-5dfd003

02 --deny-env flag to strip environment variables NEW

75

Adds `--deny-env` flag to strip named environment variables from the agent's environment before it starts, enabling use cases such as preventing OpenCode from accessing voice API keys.

— Named flag with a concrete use case but no sample invocation 2026.10.23-070444-7f0fd16

03 **Launch command write-access flag and quieter prompting** IMPROVED

75

Adds `--allow-write-state` flag to the `cplt` launch command to explicitly grant write access to the state directory, and reduces prompt noise at launch by only prompting the user when the sandbox profile has changed rather than on every invocation.

— Named flag plus described before/after prompting behaviour `2026.10.24-160000-c742e86`

04 **MCP host broker for local stdio MCP servers** NEW

65

Adds an MCP host broker enabling sandboxed agents to communicate with local stdio MCP servers, shipped first for macOS (phase 1) and then extended on Linux to run under bubblewrap (`bwrap`) namespace isolation, extending kernel-enforced sandboxing to the MCP layer.

— Names mechanism and platforms but no flag or command to invoke it

`2026.10.23-122953-428b3ee` `2026.10.23-204021-181af18`

THINNER COVERAGE BELOW

05 **Output policy controls for agent launch and exit** NEW

30

Adds output policy controls for quieter agent launch and exit behaviour.

— Thin description with no named flag or config key `2026.10.23-122953-428b3ee`

WAS THIS USEFUL? 👍 👎

containarium

Feedback ⓘ v0.116.0 NOTES ↗ RANK

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

> Containarium's heaviest window hardened its sentinel tunnel and SSH authorization stack end-to-end (TLS-by-default, self-hosted authz decider, session audit trail) while also shipping eBPF egress policy controls, audit-log retention/chain tooling, a run of in-box threat-scanning features (ClamAV, process-policy, mining detection, OpenVAS), MCP-native agent tooling, and session/run lifecycle commands.

└──▷ **WHAT SHIPPED · 29 FEATURES** most completely described first ? what's the number?

01 Audit log retention and chain-of-custody tooling NEW

95

Adds a `retention report` command and dry-run sweeper for non-chained stores to preview purges, an enforce mode that deletes expired records from those stores, a `verify-anchor` command to check a bucket's anchors against the stored chain base, a `retention.sweep.result` entry naming the chain base a purge ended on, and purging of hash-chained `audit_logs` as a prefix with a stored chain base. Also adds a GCS retention-lock anchor sink with a file-then-bucket tee for durable, tamper-resistant output.

Preview which records a retention sweep would remove from non-chained stores before any data is deleted.

```
$ containarium retention report
```

Verify a bucket's anchors against the stored chain base to confirm audit-log integrity.

```
$ containarium verify-anchor
```

— Full commands and mechanism, runnable today v0.115.0 v0.111.0

02 MCP server and agent-box tooling NEW

95

Adds the `mcp-server` binary (platform MCP) wired via `~/.claude.json` using `CONTAINARIUM_SERVER_URL` and `CONTAINARIUM_JWT_TOKEN` env vars, exposing `create_container`, `list_containers`, `expose_port`, and `list_backends` to Claude Code and Cursor. Adds the `agent-box` binary (in-box MCP) reachable over stdio via SSH, exposing `shell_exec`, `read_file`, and `write_file` for agent-native file

and shell operations inside each container. `mcp-server` now also lists its tools for a box token so MCP clients can discover tools scoped to a specific container.

Wire the platform MCP into Claude Code so an AI agent can create and manage Containarium sandboxes without leaving the editor.

```json
{
  "mcpServers": {
    "containarium": {
      "command": "/usr/local/bin/mcp-server",
      "env": {
        "CONTAINARIUM_SERVER_URL": "http://your-host:8080",
        "CONTAINARIUM_JWT_TOKEN": "<your-token>"
      }
    }
  }
}
```

Give the agent in-box shell and file access by wiring agent-box over SSH into Claude Code.

```json
{
  "mcpServers": {
    "containarium-box": {
      "command": "ssh",
      "args": ["user@your-box", "agent-box"]
    }
  }
}
```

Wire the platform MCP server into Claude Code so you can manage sandbox containers directly from your AI coding assistant.

```json
{
  "mcpServers": {
    "containarium": {
      "command": "/usr/local/bin/mcp-server",
      "env": {
        "CONTAINARIUM_SERVER_URL": "http://your-host:8080",
        "CONTAINARIUM_JWT_TOKEN": "<your-token>"
      }
    }
  }
}
```

Give an in-box agent access to shell and file operations over SSH using the `agent-box` MCP.

```json
{
  "mcpServers": {
    "containarium-box": {
      "command": "ssh",
      "args": ["user@your-box", "agent-box"]
    }
  }
}
```

Wire the platform MCP server into Claude Code so agents can manage containers without leaving the IDE.

```json
{
  "mcpServers": {
    "containarium": {
      "command": "/usr/local/bin/mcp-server",
      "env": {
        "CONTAINARIUM_SERVER_URL": "http://your-host:8080",
        "CONTAINARIUM_JWT_TOKEN": "<your-token>"
      }
    }
  }
}
```

— Exact binaries, env vars, tool names, runnable config v0.103.3

03 **TLS-hardened sentinel tunnel** BREAKING 93

Sentinel now requires TLS for tunnel sessions by default, and the legacy cleartext tunnel handshake path has been removed — any setup relying on it stops working on upgrade. Sentinel scopes certificate sync to the syncing backend's own domains and binds per-backend tunnel listeners to loopback. Tunnel sessions are TLS-wrapped on the shared sentinel listener using a pinned SPKI identity format, channel-bound token proof in tunnel handshake v2, and ALPN/SNI parsing from the peeked ClientHello. A sentinel tunnel identity bootstrap was added to `cmd/sentinel`, and a `--sentinel-pin` flag on `pool join` carries a sentinel pin into the rendered tunnel unit.

Join a pool with a sentinel pin so the rendered tunnel unit is authenticated to the sentinel identity.

```
$ containarium pool join --sentinel-pin <your-sentinel-pin>
```

— Deep mechanism, named flag, breaking note v0.103.0 v0.102.0 v0.101.2 v0.101.0

04 **Session and run termination controls** NEW 92

Adds a `--no-auto-sleep` flag and `disable_auto_sleep` config key to the CLI, API, and MCP to prevent containers from sleeping automatically. Adds a `TerminateUserSessions` RPC, a `sessions terminate` CLI subcommand, and a corresponding MCP tool to forcibly end active user sessions, plus `TERMINATE_USER_SESSIONS` scoped to an org's boxes on shared hosts. Adds a `StopBoxRun` RPC to stop a running code execution inside a box without requiring an SSH connection.

Keep a long-running CI or agent container alive without it sleeping mid-task.

```
$ containarium create --server http://your-host:8080 --no-auto-sleep
```

Terminate all active sessions for a user from the CLI — useful when revoking access immediately after an incident.

```
$ containarium sessions terminate --server http://your-host:8080
```

Wire the platform MCP (including the new terminate-sessions tool) into Claude Code so an AI agent can manage sandbox sessions.

```json
{
  "mcpServers": {
    "containarium": {
      "command": "/usr/local/bin/mcp-server",
      "env": {
        "CONTAINARIUM_SERVER_URL": "http://your-host:8080",
        "CONTAINARIUM_JWT_TOKEN": "<your-token>"
      }
    }
  }
}
```

— Multiple named RPCs/flags, runnable examples `v0.109.0` `v0.108.0` `v0.106.0` `v0.105.0`

05 **Process-policy automated threat response** NEW 90

Adds `containarium security process-policy get|set|release` CLI commands and matching `security_process_policy get|set|release` server-side tools, with an `alert|throttle|freeze` responder that supports `release` and a TTL backstop for automated threat-detect process-policy management, plus a degraded-store status line.

Inspect or tune process-policy threat thresholds — view current settings, apply a new policy, or release an active throttle/freeze on a box.

```
$ containarium security process-policy get
containarium security process-policy set
containarium security process-policy release
```

— Full CLI/API surface and responder modes named `v0.108.0`

06 **ClamAV in-box malware scanning** NEW 88

Adds a `containarium security clamav install <box>` command to install ClamAV inside a box and a tenant-run in-box ClamAV scan runner with a status enum for tracking scan state. Scan initiation is restricted to the box owner only — admin no longer bypasses this gate.

Install and run ClamAV inside a specific box to trigger an owner-gated malware scan.

```
$ containarium security clamav install <box>
```

— Command plus owner-gating mechanism, runnable `v0.108.0`

07 **SSH authorization decision mechanisms: self-hosted and cloud-forwarded**

NEW

80

Adds an `SSHAuthzService` contract with actuation mirror, `Assignment` kind, and audit fields to the proto layer; self-hosts an SSH authorization decider and `ssh_keys` registry for fully on-prem SSH authz without external dependencies, with authoritative SSH source-IP ban control, `AuthorizedKeysCommand` support on the LXC jump host for dynamic SSH key lookup, and preserved SSH authorization decisions and denied sessions for audit and review; and forwards managed SSH authorization decisions to the cloud via the platform daemon, auditing live SSH authorization with a bounded outage grace period so access decisions stay resilient during brief connectivity loss.

— Rich mechanism, no runnable command given Describes behavior, no named surface v0.104.0 v0.106.0

08 **`containarium box delete` command** NEW

75

Adds a `containarium box delete <username> --yes [--force]` subcommand for deleting a user box, with an optional `--force` flag for automated cleanup of boxes in a degraded state.

Forcibly remove a user box without an interactive prompt — useful in automated cleanup pipelines where the box may be in a degraded state.

```
$ containarium box delete alice --yes --force
```

— Exact subcommand and flags, runnable example v0.113.0

09 **Per-pool storage capacity and health reporting** NEW

73

Adds per-pool storage capacity to `containarium info` and the `get_system_info` API call, reports it in SystemInfo with gauges, logs, and built-in alerts covering peers, and exports ZFS pool fragmentation and `zpool` capacity per storage pool.

Quickly inspect how much storage each pool is consuming on your Containarium host.

```
$ containarium info --server http://your-host:8080
```

— Concrete API/command named, mechanism light v0.115.0 v0.112.0 v0.111.0

10 **SSH session audit logging and posture enforcement** NEW

73

Adds an `IngestSSHSessionRecords` RPC to ship SSH session records into the tamper-evident audit store with configurable retention, detection of a non-distro SSH listener inside a box as a new posture finding, and a box `sshd` key-only drop-in config with a posture reconciler that enforces SSH key-only authentication and marks boxes at create time. SSH authentication failures and bans are also shipped into the audit store for security event visibility.

– Named RPC/config but no CLI example `v0.109.0` `v0.101.0`

---

11 **eBPF egress policy modes and enforcement controls** NEW

70

Adds a `log_only` default egress allow-list for tenants with no stored policy so traffic is logged rather than blocked during policy gaps, plus a separate egress mode for the allow-list that gives operators independent control over enforcement and reports whether the kernel supports TCX egress enforcement. Adds per-tenant applied network-policy state reporting, an admin-only audited LOG/BLOCK switch for the baseline, an allow-list-only egress preset that restricts outbound traffic to an explicit permit list, gauges and built-in alerts for bridge DNS drift, and retry logic with readiness reporting for the guardrail/egress Postgres store installation.

– Named modes/presets but no exact flags shown `v0.116.0` `v0.115.0` `v0.111.0` `v0.105.0` `v0.101.0`

---

12 **Mining-behavior detection** NEW

65

Adds a mining-behavior detection rule over CPU samples and stratum-port flows, a standalone CPU sampler that operates when the metrics collector is disabled, and a `sentry-unavailable` finding raised when the CPU sampler is unavailable on an armed backend.

– Detection mechanism named, no operator command `v0.108.0`

---

13 **Removed legacy SSH security config fields** BREAKING

65

Removes the `SecurityConfig.require_ssh_keys` and `SecurityConfig.min_password_length` fields (dead config dropped), superseded by the new sshd key-only posture reconciler.

– Exact fields named, clear migration note `v0.101.0`

---

14 **Org-scoped model-key attribution and daemon resilience** NEW

65

Adds `on_behalf_of_org_id` to attribute a skill/crew run's model key to a specific org, an opt-in daemon self-restart when core Postgres recovers, and an option to disable the daemon-global key fallback for `key_owner` tokens.

– Named field/flag, no usage example given `v0.114.0`

---

15 **Per-box tool-access levels for platform-started runs** NEW

64

Adds per-box tool-access level recording for platform-started runs, running them as an unprivileged user once a level is recorded and applying that level to the run as CLI flags. Also records the toolchain available to a run token-free via the run wrapper, and records each proxied model call per run, returned with the run log.

— Rich mechanism, no CLI flag names given v0.115.0

THINNER COVERAGE BELOW

16 **Inbound model-gateway traffic scanning** NEW

58

Wires inbound response scanning into the daemon's model gateway so model traffic is inspected at the gateway level, reports whether a run's model traffic is scanned, caches the inbound policy read, and exports hold histograms for observability.

— Mechanism named, no operator-facing command v0.101.2

17 **Versioned Helm chart publishing** NEW

57

Versions, stamps, and publishes the Helm chart with each release — chart version `0.108.0` is available via `oci://ghcr.io/footprintai/charts/containarium-k8s`.

— Concrete chart ref, little mechanism v0.108.0

18 **Expanded client platform support** NEW

55

Publishes best-effort macOS 11 client binaries with each release, and ships a macOS ARM64 client binary (`containarium-darwin-arm64`) downloadable from GitHub releases.

Install the containarium client CLI on a macOS ARM laptop to manage sandbox containers from CI or a terminal.

```
$ curl -L -o /usr/local/bin/containarium \

https://github.com/footprintai/containarium/releases/download/v0
.101.0/containarium-darwin-arm64
chmod +x /usr/local/bin/containarium
```

— Named binary and runnable install command v0.104.0 v0.101.0

19 **Per-subject rate limits on gated deploy and policy operations** NEW 48

Adds per-subject rate limits on gated deploy and policy `Set` / `Get` operations to prevent abuse of privileged endpoints.

— Named operations, no limit numbers given v0.105.0

20 **OpenVAS per-container vulnerability scanning** NEW 46

Adds per-container opt-in OpenVAS scanning via a new OpenVAS scanner module that persists the opt-in flag per container.

— Thin one-line description, no command shown v0.101.0

21 **Coding-agent boxes with SDLC skills** NEW 43

Coding-agent boxes now ship the pinned SDLC role skills, enabling agent-native software-delivery workflows inside each sandbox, and the hosted issue-implementer shares its engineering principles with `engineer-implement`, broadening the skills surface available to coding agents.

— Prose description, no named skill list v0.103.0

22 **Proof Pack artifact** NEW 41

Introduces the Proof Pack (`proofpack`): defines the Proof Pack artifact and generates it inside the execution environment.

— Artifact named, mechanism unexplained v0.112.0

23 **Blank hold-id quarantine (Q31)** NEW 40

Quarantines blank hold IDs (Q31) during startup clamp and hold-id upgrade audits, preventing undefined-identity tasks from running.

— Named audit code, no operator action shown v0.116.0

24 **Backup-health alert signal** NEW 38

Adds an alertable signal when the `backup-health` read fails, enabling operators to detect and respond to backup-health check failures.

— Named check, no threshold or command v0.107.0

25 **Bounded-parallelism task graphs** NEW 35

Executes task graphs with bounded parallelism, enabling concurrent crew workloads without unbounded resource consumption.

— Describes behavior, no config surface named v0.116.0

26 **Per-primary auth health reporting via /status** NEW 35

Adds `/status` reporting of per-primary auth health.

— Endpoint named, no detail on fields `v0.115.0`

27 **Opt-in autoscaler scale-down timers** NEW 33

Adds opt-in autoscaler scale-down timers, controlling when the autoscaler is permitted to shrink capacity.

— Bare mention, no config key named `v0.113.0`

28 **primary_preferred metric export to Cloud Monitoring** NEW 32

Exports the `primary_preferred` metric from the sentinel to Cloud Monitoring.

— Single metric name, no usage detail `v0.103.0`

29 **Gemini turn outcome journaling** NEW 30

Journals how a Gemini turn ended, recording the finish reason and block reason.

— Bare description, no surface named `v0.108.0`

NEW DETECTION Miner-signature detection and kill-process action

NEW DETECTION exec telemetry via execwatch eBPF object

└──▷ BREAKING ON UPGRADE

! The legacy sentinel tunnel handshake path has been removed; any setup relying on the cleartext tunnel path will stop working on upgrade.

! Tunnel sessions now require TLS by default; any non-TLS tunnel session configuration will stop working on upgrade.

! The `SecurityConfig.require_ssh_keys` and `SecurityConfig.min_password_length` fields have been removed (dead config dropped).

v0.110.0 NOTES ↗ STABLE

Adds `crew plan validate`, SSH enforce routing, device-key rotation with SSHSIG proof, process guardrail status, keyless cosign signatures, and SPDX 2.3 SBOMs.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.110.0
https://github.com/FootprintAI/Containarium.git
# already have the repo? check out this version:
$ git checkout v0.110.0
```

Adds `crew plan validate` subcommand for TaskGraph proto validation.

> Adds SSH authorization enforce mode that routes every offer through itself (`ssh-authz` enforce mode).
>
> Adds device key rotation with SSHSIG proof and login expiry checks.
>
> Adds guardrail status reporting of the effective response policy via guardrail status.
>
> Adds baseline log-only response scan when no admin policy exists.
>
> +4 more

WAS THIS USEFUL? 👍 👎

◆ AI Red Teaming & Testing

NVIDIA SkillSpector

Feedback ⓘ v2.13.1 NOTES ↗

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

v2.13.1 adds atomic report and baseline writes, stdout baseline export via `-o -`, and strict exit-code-2 rejection of invalid flag combinations with `--recursive` and `--mcp-registry`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.13.1
https://github.com/NVIDIA/SkillSpector.git
# already have the repo? check out this version:
$ git checkout v2.13.1
```

Adds `skillspector baseline <path> -o -` to write a baseline as JSON to stdout.

Writes report files (`scan`, `scan --recursive`, `scan --mcp-registry`, and the batch scanner's `--output`) and baselines atomically through an already-opened, no-follow parent directory, refusing symlinked or special-file destinations.

Rejects `--use-shipped-baseline` with exit code 2 when `scan --recursive` detects child skills, before any report is written; previously the option was silently ignored.

Rejects `scan --mcp-registry` combined with `--use-shipped-baseline`, `--transitive`, `--transitive-depth`, `--transitive-allow-prefix`, or `--transitive-deny-prefix` with exit code 2, before the registry is read or a report is written; previously these options were silently ignored.

Adds `gap_fill_status: "incomplete"` with failure reasons to skill reports when the batch scanner's non-English gap-fill pass fails, and exits with code 2 instead of silently treating the pass as complete.

+1 more

NEW DETECTION MP3 HIGH finding for agent state-adoption instructions

UPDATED DETECTION P2 hidden-instruction detection boundary fix

UPDATED DETECTION TM1 repeated Python assignment analysis

└──▷ BREAKING ON UPGRADE

! `scan --recursive` with `--use-shipped-baseline` now exits with code 2 when child skills are detected, instead of silently ignoring the flag.

- ! `scan --mcp-registry` combined with `--use-shipped-baseline`, `--transitive`, `--transitive-depth`, `--transitive-allow-prefix`, or `--transitive-deny-prefix` now exits with code 2 instead of silently ignoring those options.
- ! A recursive scan whose highest score is MEDIUM (21-50) now returns CAUTION instead of SAFE, which may change automated gating decisions.

WAS THIS USEFUL? 👍 👎

iFixAi

Feedback ⓘ v4.3.0 NOTES ↗ RANK

Independent Auditing of AI Agents. Run by human or the agent itself, to answer the most crucial question in the AI Agent Economy.

> iFixAi v4.3.0 adds adapter-free chat auditing for 15 inspections, two new CLI commands (`ifixai author` and `ifixai windtunnel`), stricter evidence-floor validation before runs, V11 guilt-tripping detection, and several breaking changes to judge-budget, governance, and profile flags.

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

01 **Judge-budget accounting and runaway guard** BREAKING 85

`--judge-budget` now counts across all judge calls including retries, fallback models, and Full-mode judges, with `--dry-run` showing the budget ceiling before any probes are sent. Setting `--judge-budget 0` no longer means unlimited; it now acts as a runaway guard capped at 10x the estimate or at least 2,000 calls.

— Exact flag, numeric guard values, and behavior change specified v4.3.0

02 **Evidence-floor validation in `ifixai validate` and `run`** 85

BREAKING

`ifixai validate` now checks evidence floors and refuses fixtures that would leave an inspection without a verdict, accepting the same `-b` , `-c` , `--suite` , `--strategic` , and `--full-run` flags as `ifixai run` . `ifixai run` itself now refuses, before any call, a fixture that cannot give a selected inspection a verdict — previously such runs would proceed.

— Flags enumerated and before/after behavior given v4.3.0

03 **Structured CI exit codes** NEW 85

Adds structured CI exit codes: exit 2 when a mandatory minimum fails, exit 2 when a selected mandatory minimum is set aside unchecked, and `overall.passed` in JSON is true exactly when the exit code is 0.

— Exact exit codes and JSON field named v4.3.0

04 **`ifixai windtunnel` defect-injection command** NEW 83

Adds `ifixai windtunnel` command with `controls`, `run`, `sweep`, and `sensitivity` subcommands to inject a single governance defect and measure whether each inspection catches it plus its false-positive rate; it runs offline without affecting grades.

— Full subcommand list and mechanism, runnable command v4.3.0

05 **Removed legacy flags and components** BREAKING 78

`--governance` is now ignored; governance rules must be placed in the fixture's `governance:` block. `--profile` was removed and now errors; use `--mode standard` or `--mode full` instead. The record/replay bridge, the unused embedding classifier, and the `ifixai.examples` scripts have been removed.

— Exact flags, migration path, and removed components named v4.3.0

06 **Adapter-free chat auditing for 15 inspections** BREAKING 75

iFixAi can now audit any agent over a plain chat endpoint or bare model API without needing an adapter, covering inspections `B01`, `B02`, `B04`, `B23`, `C02`, `C05`, `C11`, `M02`, `M03`, `M07`, `P01`, `P08`, `V08`, `X04`, and `X11`. Because these checks are now decided by the agent's chat replies rather than its hooks, scores for them cannot be compared with 4.2.x runs.

— Names all 15 inspection codes but gives no runnable command v4.3.0

07 **`ifixai author` fixture generator command** NEW 72

Adds `ifixai author` command to generate a fixture from a short JSON agent profile; `--example` writes a sample profile and `--force` allows overwriting an existing file.

— Exact command and two flags, directly runnable v4.3.0

08 **`not_applicable` field for skipped checks** NEW 70

Adds a `not_applicable` field in reports explaining why a check was set aside before scoring, and the Python API's `run_single` now raises `ValueError` with the reason when a check is not applicable.

— Named report field and API behavior, usable in code v4.3.0

09 **V11 Guilt Tripping exploratory detection** NEW 63

Adds V11 Guilt Tripping exploratory detection: 40 four-turn companion chats verify the agent never encourages self-harm, demeans a struggling user, or fails to route to a human by the turn the risk appears.

— Scenario and counts specified but no runnable entry point v4.3.0

10 **Optional `--api-key` with env var fallback** IMPROVED 60

Makes `--api-key` optional; the key falls back to the provider's environment variable (e.g. `OPENAI_API_KEY`) when the flag is omitted.

— Exact flag and env var example given v4.3.0

THINNER COVERAGE BELOW

11 **Default MiniMax model updated** IMPROVED 35

Sets the default MiniMax model to `MiniMax-M3.1-Flash`.

— Single config default named, no further detail v4.3.0

└──▷ BREAKING ON UPGRADE

! Scores for B01, B02, B04, B23, C02, C05, C11, M02, M03, M07, P01, P08, V08, X04, and X11 are now decided by the agent's replies over chat rather than its hooks and cannot be compared with 4.2.x runs.

! `ifixai run` refuses, before any call, a fixture that cannot give a selected inspection a verdict — previously such runs would proceed.

! `--judge-budget 0` is now a runaway guard (10x the estimate, at least 2,000 calls) instead of unlimited.

! `--governance` is ignored; governance rules must be placed in the fixture's `governance:` block.

! `--profile` was removed and now gives an error; use `--mode standard` or `--mode full` instead.

! The record/replay bridge, the unused embedding classifier, and the `ifixai.examples` scripts have been removed.

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Frontier Models

Anthropic

Feedback ⓘ snapshot-20261025 NOTES ↗ RANK

The Claude Developer Platform is the API for building applications on Anthropic's Claude models that generate, analyze, and automate work.

> Anthropic introduced a beta Claude Tag API that lets Claude Enterprise organizations manage Slack channel and workspace tagging settings directly via the Claude Developer Platform.

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

01 **Claude Tag API (beta) for Slack settings** NEW 83

Adds the Claude Tag API in beta, accessible via the `experimental-claude-tag-2026-10-21` beta header, for Claude Enterprise organizations. New `/v1/organizations/claude_tag/` endpoints let callers read and modify Claude Tag settings on Slack channels and Slack workspaces, inspect which settings apply in a given channel and their sources, and attach Access bundles. Access requires an Admin API key created in `claude.ai` with `read:claude_tag` or `write:claude_tag` scope, issued by a primary owner, and the API is not available to Claude Platform (Claude Console) organizations.

— Names endpoint path, beta header, and exact scopes; lacks usage example.

snapshot-20261025

WAS THIS USEFUL? 👍 👎

◆ Open-Weight Models

Liquid AI LFM2.5-2.6B (Liquid AI)

Feedback ⓘ

new snapshot · Oct 25, 2026 MODEL ↗

New snapshot 20260811 listed on OpenRouter (liquid/lfm-2.5-2.6b), first seen 25 Oct 2026.

WAS THIS USEFUL? 👍 👎

DeepSeek V4 Flash 0731 Feedback

price cut · Oct 24, 2026 MODEL ↗

DeepSeek V4 Flash 0731 is a chat model from DeepSeek.

Price cut on OpenRouter (deepseek/deepseek-v4-flash-0731): $0.44 → $0.0444 in, $1.32 → $0.1333 out per million tokens (-90%), first seen 24 Oct 2026.

WAS THIS USEFUL?

DeepSeek V4.1 Flash Feedback price cut · Oct 24, 2026

MODEL ↗

DeepSeek V4.1 Flash is a multimodal model from DeepSeek.

Price cut on OpenRouter (deepseek/deepseek-v4.1-flash): $0.30 → $0.02 in, $1.20 → $1.00 out per million tokens (-93%), first seen 24 Oct 2026.

WAS THIS USEFUL?

Z.ai GLM 5.3 Flash Feedback price increase · Oct 24, 2026

MODEL ↗

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

Price increase on OpenRouter (z-ai/glm-5.3-flash): $0.0052 → $0.15 in, $0.1985 → $0.50 out per million tokens (+2785%), first seen 24 Oct 2026.

WAS THIS USEFUL?

Google Gemma 4 26B A4B

Feedback

price cut · Oct 24, 2026

MODEL ↗

Gemma 4 26B A4B is a multimodal model from Google.

Price cut on OpenRouter (google/gemma-4-26b-a4b-it): $0.09 → $0.0675 in, $0.30 → $0.225 out per million tokens (-25%), first seen 24 Oct 2026.

WAS THIS USEFUL?

Moonshot AI Kimi K3

Feedback

price cut · Oct 24, 2026

MODEL ↗

Kimi K3 is a multimodal model from Moonshot AI.

Price cut on OpenRouter (moonshotai/kimi-k3): $0.239 → $0.239 in, $11.936 → $10.80 out per million tokens (-10%), first seen 24 Oct 2026.

WAS THIS USEFUL?

Moonshot AI Kimi K3.1 Pro

Feedback

released · Oct 24, 2026 MODEL ↗ BLOG ↗

Moonshot AI: "Introducing Kimi K3.1 Pro, our most capable 3T-class open-weight model with 1M-token context and major advances in coding, 3D modeling, and financial research."

Moonshot AI says: "We are also releasing Kimi K3.1 Pro Highspeed , which delivers 2-3× the speed of Kimi K3.1 Pro for more responsive interactions."

Moonshot AI says: "On Terminal-Bench 4.0 , which evaluates coding agents on realistic, resource-constrained terminal tasks using strict task-specific tests, Kimi K3.1 Pro scores 65.7%, ahead of GPT-6 Astra (58.2%) and narrowly behind Claude Opus 5.5 (66.4%) ."

Moonshot AI says: "And on Kimi Code Bench 3.0 Long-horizon , our internal benchmark that tests coding agents' multi-step execution, tool use, and end-to-end problem solving across diverse engineering domains, Kimi K3.1 Pro scores 61.2% , above K3's 55.95% ."

Moonshot AI says: "On BenchCAD , which tests 3D reconstruction from multi-view renderings through CAD code generation, Kimi K3.1 Pro scores 92.7% with tools , approaching the performance of Claude Opus 5.5 (96.2%) and GPT-6 Astra (95.9%) ."

Moonshot AI says: "Kimi K3.1 Pro achieves about 4.2× K3's performance."

+2 more

WAS THIS USEFUL? 👍 👎

sky7350 Mica-v0.5-4B

Feedback released · Oct 24, 2026

MODEL ↗

Mica-v0.5-4B from sky7350 was released on 27 Sep 2026. Open weights on Hugging Face.

WAS THIS USEFUL? 👍 👎

NVIDIA-Nemotron-3.5-Bolt-E4B-4GB

Feedback

released · Oct 24, 2026 MODEL ↗

NVIDIA-Nemotron-3.5-Bolt-E4B-4GB from NVIDIA was released on 5 Oct 2026. Open weights on Hugging Face.

WAS THIS USEFUL? 👍 👎

NVIDIA-Nemotron-3.5-Bolt-E4B-4GB-DFlash

Feedback

released · Oct 24, 2026 MODEL ↗

NVIDIA-Nemotron-3.5-Bolt-E4B-4GB-DFlash from NVIDIA was released on 5 Oct 2026. Open weights on Hugging Face.

WAS THIS USEFUL? 👍 👎

ConvAI Innovations Laya

Feedback ⓘ v0.4.14 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 v0.4.14 release rounds out the Go SDK with calibration support, email utilities, routing, and hook-based extensibility, alongside new runnable samples.

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

01 **Go SDK expansion: calibration, email, routing, hooks, samples**

NEW 68

The Go SDK gains calibration payload installation at `Open`, the `laya.email` module (`CleanBody`, `State`), the routing half of `laya.router` (`Route`, `RouteBatch`), `on_route` hooks with a hook registry on the router, and runnable quickstart and benchmark samples.

— Names every ported function and hook, but no usage shown v0.4.14

WAS THIS USEFUL? 👍 👎

Inference.net

Feedback

MiniMax M3.1 Flash is live on Inference: Day 0 · NOTES ↗ · RANK

Inference.net is a inference infrastructure for AI-native teams. Run, monitor, and optimize production AI workloads with lower cost, faster latency, and dedicated support from Inference.net.

> Inference.net added MiniMax M3.1 Flash, a large Mixture-of-Experts model, to its serverless API.

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

01 **MiniMax M3.1 Flash on serverless API** NEW 60

Inference.net now serves MiniMax M3.1 Flash, a 428B-parameter Mixture-of-Experts model optimized for coding, agents, and complex workflows, via its serverless API at $0.10 / $0.50 per million tokens (input/output).

— Names model, size, pricing, and API access, but no setup steps.

MiniMax M3.1 Flash is live on Inference…

WAS THIS USEFUL? 👍 👎

Mistral AI Mistral Small 3.2 24B

Feedback

price cut · Oct 23, 2026 · MODEL ↗

Mistral Small 3.2 24B is a multimodal model from Mistral AI.

Price cut on OpenRouter (mistralai/mistral-small-3.2-24b-instruct): $0.0938 → $0.075 in, $0.25 → $0.20 out per million tokens (-20%), first seen 23 Oct 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3.5-9B

Feedback

price increase · Oct 23, 2026

MODEL ↗

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

Price increase on OpenRouter (qwen/qwen3.5-9b): $0.08 → $0.17 in, $0.13 → $0.25 out per million tokens (+112%), first seen 23 Oct 2026.

WAS THIS USEFUL?

Alibaba Qwen3.8 27B

Feedback

price increase · Oct 23, 2026

MODEL ↗

Qwen3.8 27B is a multimodal model from Alibaba.

Price increase on OpenRouter (qwen/qwen3.8-27b): $0.04 → $0.20 in, $1.35 → $2.50 out per million tokens (+400%), first seen 23 Oct 2026.

WAS THIS USEFUL?

Arcee AI Trinity Large Thinking

Feedback

retirement scheduled · Oct 23, 2026

MODEL ↗

Trinity Large Thinking is a reasoning model from Arcee AI.

OpenRouter lists a retirement date of 26 Oct 2026 for arcee-ai/trinity-large-thinking, first seen 23 Oct 2026.

WAS THIS USEFUL?

OTHER / UNCATEGORIZED

◆ UNCATEGORIZED

Foremerge

Feedback ⓘ v0.6.0 NOTES ↗ RANK

Foremerge is a coordination protocol for coding agents that detects intent conflicts before code conflicts.

> Foremerge v0.6.0 makes ledger schema migration an explicit, backed-up operation, adds a way to close intents that landed outside the normal acceptance flow, extends the MCP server to a new protocol revision, and adds several response fields and validation warnings for conflict tracking.

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

01 Explicit ledger migration via `foremerge ledger migrate`

93

BREAKING

Migration is no longer a side effect of opening a ledger: every command and MCP server now refuses a ledger at an older schema with `MIGRATION_REQUIRED` instead of migrating it automatically. The new `foremerge ledger migrate` command performs the migration, with `--yes` to apply, an automatic backup to `backups/<timestamp>-schema<N>-before-migration/`, an integrity check, and rollback instructions; ledgers last opened by 0.4.x or 0.5.x need no migration, only those last opened by 0.3.x do. The `foremerge::Store::open` library API likewise no longer migrates older ledgers and refuses them with `MIGRATION_REQUIRED`, and a build whose ledger schema is newer than any published release refuses `ledger migrate` and `ledger reset` unless `--allow-development-build` is passed.

Safely migrate a ledger last opened by Foremerge 0.3.x — preview the migration first, then apply it.

```
$ foremerge ledger migrate
foremerge ledger migrate --yes
```

— Exact commands, flags, backup path and error code given v0.6.0

02 `work discard --landed` closes intents merged outside acceptance

NEW

86

`foremerge work discard --landed REF` (MCP/HTTP field `landed_ref`) closes an intent whose work reached Git via squash or rebase merge without going through the normal acceptance flow. It refuses with `TARGET_DIVERGED` if the commit cannot be

verified, or `STALE_CHANGESET` if the ChangeSet was published with uncommitted changes.

Close an intent whose branch was squash-merged by a maintainer without going through the normal acceptance flow.

```
$ foremerge work discard --landed main
```

— Exact flag, field name and both error codes given v0.6.0

03 **MCP protocol revision 2026-07-28 support** IMPROVED 75

`foremerge mcp` now speaks MCP protocol revision `2026-07-28` alongside `2025-11-25`, negotiated per-request via `io.modelcontextprotocol/protocolVersion` in `_meta`. `server/discover` returns both supported versions, capabilities, instructions, server identity, and `resultType` on every result; `tools/list` and `server/discover` carry `ttlMs` and `cacheScope` caching hints; `ping` and `initialize` answer `-32601`; missing client capabilities answer `-32602`; and an unknown protocol version answers `-32022` with the supported list.

— Detailed protocol mechanism and error codes, no runnable example v0.6.0

04 **New conflict-state metadata fields on responses** NEW 67

`check_conflicts` responses now include an `ephemeral` field, `true` for in-memory findings with per-call `eph_` ids that cannot be stored, `false` for stored conflict reads of a published intent. An `as_of_event` field (the ledger's latest event sequence number) is now added to every open-conflict list returned by `check_conflicts`, `start_work`, `publish_changeset`, `get_intent`, and `status`, letting two reads with the same value be confirmed to reflect the same conflict state. `claim_work` responses gained a separate `scope_warnings` field to surface scope-key canonicalization warnings apart from the existing `warnings` field, which holds conflicts.

— Field names and endpoints given, but no usage example v0.6.0

05 **New validation warnings in publish_intent and claim_work** NEW 62

`publish_intent` now warns when an intent declares no scopes, since it can then only be matched by summary similarity and can never produce a HIGH finding. `publish_intent` and `claim_work` also warn when a `symbol:` key uses dot notation (e.g. `CheckoutController.checkout`) that canonicalizes incorrectly, naming the alias and suggesting the `Class::method` spelling instead.

— Names the two warnings and the fix, no example shown v0.6.0

└──▷ **BREAKING ON UPGRADE**

! Every command and MCP server now refuses a ledger at an older schema with `MIGRATION_REQUIRED` instead of migrating automatically — run `foremerge ledger migrate` (then `--yes`) to migrate; ledgers last opened by 0.4.x or 0.5.x need no migration, only ledgers last opened by 0.3.x do.

! The `foremerge::Store::open` library API no longer migrates older ledgers; it refuses them with `MIGRATION_REQUIRED`. Migration is only available through `foremerge ledger migrate`.

! A build whose ledger schema is newer than any published release refuses `ledger migrate` and `ledger reset` unless `--allow-development-build` is passed.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · October 25, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue