THE AI TOOLCHAIN

NO. 047

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED OCTOBER 5, 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 99 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 22 tools

▾ // THE BRIEF HIDE

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

Three unrelated things stand out. claude-mem cut SessionStart from up to 44 seconds to milliseconds, so agent context loading stops being ble wait. vLLM v0.31.0 caches weights in a daemon so a restarted

vim ? help

engine skips reloading them from disk. OpenRouter now runs shell commands in its own sandbox, so no caller provisions a container.

DEPLOY

Restart an inference engine without paying the weight-load cost again

vLLM's weight-cache daemon keeps model weights resident across engine restarts, turning a multi-minute reload into a fast bring-up during config changes or crash recovery. SGLang unifies Mamba and KV cache into one page-granular buffer, so hybrid-attention models stop reserving two separate memory pools.

vLLM · SGLang

BUILD

Give a model a shell without standing up sandbox infrastructure

Server-side code execution runs the commands inside OpenRouter's sandbox, so teams that wanted a code-interpreter loop no longer have to build container provisioning, lifecycle and isolation themselves before the first call works.

OpenRouter

OPERATE

Stop waiting on agent memory at session start

claude-mem's indexed queries and local-first context building take SessionStart from up to 44 seconds to milliseconds — the difference between an agent that feels dead and one that answers. LangSmith's `/v2/threads/stats` returns per-project latency, token and cost in one call, so dashboards stop being assembled by hand from trace exports.

claude-mem · LangSmith

GOVERN

Put a hard boundary around what an agent is allowed to spend or touch

NEXUS's Sovereign Payment Gateway is fail-closed: a payment that doesn't clear all ten control components doesn't execute, rather than executing and being audited later. Letta Code's read-only file access means a subagent can read a repo without the risk that it writes to it.

ai-safe2-framework

OFFENSIVE SECURITY

◆ Exploitation & C2

xalgorix

Feedback ⓘ v4.6.145 NOTES ↗ RANK

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

> xalgorix v4.6.145 adds a bounded regex denial-of-service detection task to its assessment pipeline and introduces parameterized route deduplication for smarter route discovery.

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

01 Bounded regex denial-of-service detection task NEW

46

Assessments now include a bounded regex denial-of-service detection task that runs after reconnaissance and probe validation, when the relevant testing phase is selected.

— Names the task and its place in the pipeline but no further mechanism. v4.6.145

02 Parameterized route deduplication in route discovery IMPROVED

18

Route discovery is described as smarter, with parameterized route deduplication, but no further detail on how it works is given.

— Mentioned only in passing with no mechanism or scope. v4.6.145

WAS THIS USEFUL? 👍 👎

BUILD

◆ AI Coding Agents

yetone magpie

Feedback ⓘ v0.1.985 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 added a full TUI alongside richer gateway controls — environment-variable bridging, per-CLI model switching, intent-based routing, and smart-failover routing groups across 24+ provider presets — plus deeper provider management: automatic balance-field detection, Cursor Private Inference support, per-model compact thresholds and pricing, upstream-aggregator visibility, automatic gateway port takeover, and label suffix controls.

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

01 **Per-CLI model switching subcommands** NEW 90

Agents can be switched directly from the terminal with direct subcommands such as `magpie claude <provider/model>` and `magpie codex <provider/model>`, which rewrite that agent's own config file safely and atomically — e.g. `magpie claude deepseek/deepseek-v4-pro` and `magpie codex moonshot/kimi-k2.5`.

Switch Claude Code to DeepSeek and Codex to Kimi from the terminal without touching config files manually.

```
$ magpie claude deepseek/deepseek-v4-pro
magpie codex moonshot/kimi-k2.5
```

— Runnable commands with config-rewrite mechanism named product docs

02 **Gateway port takeover from stale instances** IMPROVED

88

The Gateway card now detects an older magpie instance holding the gateway port, quits it, and takes over serving the gateway itself, replacing prior behavior that only reported the conflicting process. Takeover first verifies via `proc.IsMagpie` that the program path is actually a magpie binary before terminating it; other programs holding the port are reported but never touched. The takeover result is structured JSON with a `reason` field (`not-magpie` , `unseen` , `denied` , `stuck` , `respawned` , `taken` , `other` , `error`) and a `pid` field identifying the conflicting process, and Linux port-listener resolution now works via `/proc/net/tcp` and `/proc/net/tcp6` without elevated privileges.

— Full before/after behaviour with exact internal fields v0.1.977

03 **Gateway base-URL environment variables** NEW

85

magpie exposes `OPENAI_BASE_URL` , `ANTHROPIC_BASE_URL` , and `GOOGLE_GEMINI_BASE_URL` environment variables so any OpenAI-, Anthropic-, or Gemini-compatible agent can be pointed at the local magpie gateway, e.g. `export OPENAI_BASE_URL=http://127.0.0.1:3425/v1 OPENAI_API_KEY=magpie` .

Point any OpenAI-compatible agent at the magpie gateway to gain access to all configured providers without changing the agent's own provider logic.

```
$ export OPENAI_BASE_URL=http://127.0.0.1:3425/v1
OPENAI_API_KEY=magpie
```

— Exact env vars and a runnable export command product docs

04 **Automatic balance field detection for custom providers** NEW

85

When a custom provider's Balance field is left empty, magpie auto-detects the correct balance field path, covering the `new-api` / `one-api` key-usage endpoint (`/api/usage/token` → `$data.total_available / 500000`), the sub2api profile endpoint (`/api/v1/user/profile` → `$data.balance`), and the OpenAI legacy credit-grants endpoint (`/v1/dashboard/billing/credit_grants` or `/dashboard/billing/credit_grants` → `$total_available`). The Balance URL editor's Balance field input placeholder now updates live as you type a URL to preview the path that will be used.

— Three endpoints and exact field paths named verbatim v0.1.985

05 **Per-model and per-provider compact thresholds** NEW

85

`settings.ModelCompacts` accepts keys in `provider/model` or `provider/*` form to set per-model or per-provider compact thresholds for Codex and Claude Code long conversations; `SetModelCompacts` saves these, with the smallest threshold winning for model groups. A global compact threshold is also configurable in Settings → Long conversations, accepting any token count or `Full window`.

— Config key, API name, and exact UI path given `v0.1.979`

06 **Per-model price editing** NEW 85

The provider editor's `Names & levels` panel gains per-model price editing alongside the model name, showing the vendor list price greyed out until a custom price is entered — mirroring what `magpie model price` sets. `Input`, `Output`, `Cache read`, and `Cache write` price fields (in US dollars per million tokens) are editable per model, with `OwnPrice: true` restoring the vendor list price.

— Exact UI panel, fields, and CLI counterpart named `v0.1.971`

07 **Routing groups with smart failover** NEW 80

`magpie group add <name> models=<model1>,<model2> routing=smart` creates a routing group that automatically continues on another model if one hits a quota limit; agents can then be pointed at it, e.g. `magpie claude group/opus-anywhere`, as demonstrated with a two-subscription Claude Opus failover group.

Create a smart failover routing group across two Claude subscriptions so your agent automatically continues if one hits a quota limit.

```
$ magpie group add "Opus anywhere" models=claude/claude-opus-5-5,copilot/claude-opus-5.5 routing=smart
magpie claude group/opus-anywhere
```

— Full runnable command sequence with flags `product docs`

08 **Upstream provider visibility behind aggregators** NEW 80

Adds an `upstream` field (JSON key `upstream`) to `Route` and `Try` trace objects, exposing which provider an aggregator such as OpenRouter actually used to answer a request (e.g. DeepInfra, StreamLake, Novita); the gateway parses the `provider` key from OpenRouter Chat, Anthropic Messages, and streaming responses to populate it, covering both JSON bodies and stream chunks. The GUI surfaces this as `Upstream: DeepInfra` alongside the routed provider.

— Named trace fields and parsing sources, no CLI step `v0.1.968`

09 **Cursor Private Inference detection in gateway** NEW 75

The gateway detects a Cursor User-Agent header (e.g. `Cursor-CLI/…` or `Cursor/3.x`) and routes the request to the `cursor-local` agent, so Cursor Private Inference using a non-Cursor key still receives its own model list with `supports_reasoning` populated in model `capabilities`.

– Clear detection rule and resulting behaviour named v0.1.984

10 **Upstream pinning for Cline plugin provider** IMPROVED 75

The `pinUpstream` (Upstream) toggle, previously only available on ClinePass providers, now extends to the Cline plugin provider (`@magpie-community/opencode-cline-auth`), so DeepSeek model requests from the plugin route directly to DeepSeek's own API and keep their prompt cache.

– Named plugin, toggle, and resulting routing behaviour v0.1.959

11 **Suffix control and live label preview for names** IMPROVED 70

A `SuffixOwn` suffix mode (set via SetSuffixMode(SuffixOwn)) drops the provider or `· routing group` suffix from names the user has explicitly set, while keeping suffixes on vendor-discovered models and ambiguous names to avoid duplicates in agents' lists. The GUI model-name editor and routing-group editor both gained a live Agents lists show …' / In agents lists' preview with a suffix toggle, fixing previously truncated long group names (e.g. `· rou…`) in narrow agent menus.

– Named API call and UI previews, exact flag given v0.1.967

THINNER COVERAGE BELOW

12 **TUI subcommand for agents and routing** NEW 55

`magpie tui` launches a full terminal UI for managing agents, providers, and routing, alongside the existing CLI and menu-bar app.

– Names exact command but little on internal behaviour product docs

13 **Provider presets across 24+ vendors** NEW 55

Ships built-in presets for Anthropic, OpenAI, Google Gemini, DeepSeek, Kimi, Zhipu GLM, MiniMax, StepFun, Qwen, Baidu Qianfan, Tencent Cloud, Huawei Cloud MaaS, Volcengine Ark, Mistral, Groq, xAI, OpenRouter, Together, Fireworks, SiliconFlow, NVIDIA NIM, ModelScope, Ollama, LM Studio, plus any OpenAI-compatible or Anthropic-compatible URL.

– Long named list but no setup steps given product docs

14 **Intent-based model routing** NEW 45

A configurable small model reads each conversation turn and routes it to different models based on request type — for example sending tests to a strong model and quick

questions to a fast model.

— Describes mechanism but no config surface named `product docs`

---

15 **Quota reset reminder** NEW 45

`settings.ResetReminder` notifies before a long quota window renews or reset credits expire while much of the window remains unused.

— Names the setting but no trigger timing specifics `v0.1.971`

---

16 **Claude subscription bridge and MCP tool routing** NEW 35

Bridges Claude subscription sign-in to the local `claude` binary and routes agent tools over MCP.

— One-line description with no mechanism detail `product docs`

WAS THIS USEFUL?  

claude-mem

Feedback ⓘ v13.31.0 NOTES ↗ RANK

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

> claude-mem's biggest change this window is a major SessionStart latency overhaul — from up to 44 seconds down to milliseconds — via new indexed queries, local-first context building, and caching, alongside an iFlytek Spark provider preset, richer OpenRouter observer streaming/metadata, a worker readiness endpoint, longer-running knowledge-base streaming, and Supabase Realtime cloud sync for cmem.ai Pro.

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

01 SessionStart latency cut from 44s to milliseconds IMPROVED

83

Schema v63 adds indexes keyed on project and date so SessionStart fetches only the newest memory rows per project instead of scanning the full table, cutting typical reads from ~21,700 rows (63 MB) and 0.8–1.7s down to ~4ms. Schema v64 adds covering indexes for project-alias lookups, reducing pages read from 7,499 to 355 and lookup time to ~9ms. SessionStart context is now built from the local database immediately, with cloud-sync hub pulls running in the background so a slow or unreachable hub never delays a session, and the `claude-mem-cowork` hook skips the cmem.ai cloud read at session start and for agent prompts when a local claude-mem hook is already installed. The worker also pre-caches SessionStart context in `~/.claude-mem/state/context-cache/`, refreshing ~2 seconds after any change so new sessions read a file instead of waiting up to 1.5 seconds for a cloud pull (falling back to the live path if the cache is missing or over 24 hours stale), and hooks now write events to `~/.claude-mem/state/hook-spool/` and return immediately without waiting on the worker, with unprocessed files older than 7 days moved to `hook-spool/expired/`.

— Mechanism and before/after numbers fully given; paths not directly user-invoked

v13.31.0 v13.30.0

02 Worker startup phase streaming at GET /api/ready NEW

80

Worker startup phases (`starting`, `db_ready`, `routes_ready`, `ready`, or `failed`) are now streamed at `GET /api/ready`; `CLAUDE_MEM_WEDGED_WORKER_UPTIME_S` now applies only to the launcher's port reclaim.

Check whether the worker is fully ready (all startup phases complete) before scripting hook setup or health checks.

```
$ curl -s GET http://localhost:<port>/api/ready
```

— Exact endpoint, phase values, and a runnable curl example v13.30.0

03 **OpenRouter observer request metadata and streaming** IMPROVED

78

OpenRouter observer requests now include a `session_id` (SHA-256 hash of the Claude Code session id), a `trace_id`, `trace_name` `claude-mem observer`, `generation_name` (one of `init`, `observation`, `summary`, `field_compression`, or `telegram_wrapup`), and `claude_mem_version`, for cache-affinity routing and PostHog mapping via OpenRouter Broadcast. OpenRouter and OpenAI-compatible observer requests now also stream, keeping a reply alive with a 90-second idle limit and 300-second cap; `CLAUDE_MEM_LLM_TIMEOUT_MS` now applies only to non-streaming providers (Gemini, Codex, cmem.ai gateway, or endpoints that reject streaming).

— Concrete fields and env var named but no runnable example given v13.30.0

04 **Cloud sync via Supabase Realtime for cmem.ai Pro** IMPROVED

66

Cloud sync for cmem.ai Pro now receives live updates via a private Supabase Realtime channel instead of the sync hub WebSocket; set `CLAUDE_MEM_CLOUD_SYNC_WS=false` to turn live updates off.

— Exact env var to toggle the behavior is given v13.30.0

THINNER COVERAGE BELOW

05 **Streaming for long knowledge-base prime and query calls**

IMPROVED

55

Long knowledge-base `prime` and `query` calls now stream with a 10-second heartbeat, allowing up to 15 minutes of runtime instead of hitting a 30-second deadline.

— Names the calls and limits but no invocation detail v13.30.0

06 **iFlytek Spark provider preset** NEW

50

Adds iFlytek Spark as an OpenAI-compatible provider preset (Astron MaaS), with default model `spark-x2.5`.

— Names the preset and default model but no setup steps v13.31.0

WAS THIS USEFUL? 👍 👎

Diagram Design

Feedback ⓘ changes since 2026-09-05 CODE ↗

RANK

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

> Diagram Design's v2.5.10 release adds three new layout grammars — Architecture Delta, Exploded Axonometric, and Axonometric Plan — expanding the kinds of editorial diagrams the skill can generate.

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

01 **Architecture Delta layout grammar** NEW 70

Adds `type-architecture-delta` layout grammar for comparing synchronized topologies through a Before · Changes · After ledger of added, removed, changed, moved, and rewired objects, documented in `references/type-architecture-delta.md`.

— Names mechanism and reference doc but no runnable command. product docs

02 **Exploded axonometric layout grammar** NEW 70

Adds `type-exploded` layout grammar for drawing objects in 2:1 dimetric projection with parts lifted apart at equal gaps, including animated explode support, documented in `references/type-exploded.md`.

— Names projection mechanism and reference doc, no exact usage command. product docs

03 **Axonometric plan layout grammar** NEW 70

Adds `type-axonometric-plan` layout grammar for rendering floor plans or campus sites in axonometric projection, with walls cut at desk height and phased animation support, documented in `references/type-axonometric-plan.md`.

— Describes rendering behaviour and reference doc, lacks concrete usage steps.

product docs

changes since 2026-09-05 CODE ↗

Adds waterfall chart, heatmap, Streamgraph, Marimekko, lifecycle phase map, traceable block decomposition, Excalidraw import, and a print-A3-landscape preset.

└──▷ **GET THIS VERSION**

```
$ git clone --branch commits-2026-09-05
https://github.com/cathrynlavery/diagram-design.git
# already have the repo? check out this version:
$ git checkout commits-2026-09-05
```

Adds `print-a3-landscape` size preset to the output spec for print-ready diagram export.

Adds waterfall chart as a new visual type.

Adds heatmap as visual type #40.

Adds Marimekko as a Treemap variant.

Adds Streamgraph as a Line variant.

+5 more

WAS THIS USEFUL? 👍 👎

Letta Code

Feedback ⓘ v0.34.3 NOTES ↗ RANK

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

> Letta Code v0.34.3 adds read-only file access controls, deeper subagent nesting with parent-child messaging and MCP server inheritance, and merges model settings instead of replacing them.

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

01 **Merge behavior for `letta model set`** IMPROVED 80

The `letta model set` command now merges `model_settings` rather than replacing them wholesale, so partial updates no longer wipe out existing configuration.

— Names exact command and config field, but no example of merged keys v0.34.3

THINNER COVERAGE BELOW

02 **Read-only file policy in MemFS** NEW 55

MemFS gains a read-only file policy that restricts agent write access to the filesystem.

— Explains purpose but no config key or activation steps given v0.34.3

03 **Nested subagents with parent messaging** NEW 50

Subagents can now launch one additional nested layer deep and message their parent agent directly.

— States nesting depth and messaging behavior but no invocation syntax v0.34.3

04 **Automatic MCP server inheritance for subagents** IMPROVED 45

Subagents now automatically inherit parent MCP servers without requiring explicit configuration.

— Describes behavior change but no setup or override details v0.34.3

WAS THIS USEFUL? 👍 👎

Token Optimizer

Feedback ⓘ v5.13.34 NOTES ↗ RANK

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

> Token Optimizer added support for the OpenCode 2 plugin system, running it alongside the existing OpenCode plugin system.

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

01 **OpenCode 2 plugin system support** NEW 35

Token Optimizer now supports the OpenCode 2 plugin system, running alongside the existing OpenCode plugin system simultaneously.

— Names the integration but no mechanism or config details v5.13.34

WAS THIS USEFUL? 👍 👎

Graphify Labs Graphify

Feedback ⓘ v0.9.76 NOTES ↗

RANK

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

> Graphify v0.9.76 broadens type extraction to four more languages and patches a path-traversal vulnerability in its CLI hook installer.

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

01 **Type extraction for Zig, C++, VB.NET, and Pascal** NEW

70

Adds type extraction for Zig tagged-union variants (union(enum)), emitting a node per variant with a `case_of` edge; C++ `union` specifiers as class-like type nodes with their members; VB.NET enum members, now linked via `case_of` instead of `contains`; and Pascal enumerated types and their values.

– Names exact languages, node types and edge kinds, no commands given v0.9.76

THINNER COVERAGE BELOW

02 **Path-traversal fix in hook installer** IMPROVED

35

Fixes a path-traversal vulnerability in `graphify hook install`.

– Names the fixed command but no detail on the vulnerability mechanism v0.9.76

WAS THIS USEFUL? 👍 👎

Claudoscope

Feedback ⓘ v1.3.0 NOTES ↗

Claudoscope is a native macOS menu-bar app that tracks Claude Code sessions, estimates costs, scans secrets, and surfaces analytics.

Fleet view brings multi-agent oversight via `~/.claude/sessions/`, with MCP tools `list_agents` / `get_agent`, per-agent budgets, crash detection, and cold-cache cost telemetry.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v1.3.0
https://github.com/cordwainersmith/Claudoscope.git
# already have the repo? check out this version:
$ git checkout v1.3.0
```

Adds Fleet view: a full-width board across all Claude Code sessions from `~/.claude/sessions/` active now or in the last 24 hours, with `Needs you`, Working, and Parked lanes, oldest-first wait queue, and a skipped-permissions toggle.

Adds `list_agents` and `get_agent` MCP server tools returning Fleet board agents, states, wait reasons, process details, cost, and context — with agents needing attention first.

Adds per-agent dollar budgets in the Fleet inspector: crossing a budget (and each doubling) posts a notification; a budget chip turns amber at 80% and red at 100%.

Adds crash detection: when a process disappears from Claude Code's registry while reported busy with no Stop hook, its card lands as Failed with `Process exited mid-turn`.

Adds response-time and cold-cache cost telemetry: mean and longest wait today, cold-cache answer count, estimated re-cache cost, stored locally for 30 days.

+3 more

NEW COMPLIANCE FRAMEWORK CFG023 — OpenTelemetry export variables in project settings

NEW COMPLIANCE FRAMEWORK CFG024 — reserved MCP server name `anthropic-skills`

NEW COMPLIANCE FRAMEWORK CFG025 — MCP entry of type `sdk`

NEW COMPLIANCE FRAMEWORK CFG026 — managed-only keys in user settings

NEW COMPLIANCE FRAMEWORK HOOK005 — agent-type hook on PermissionRequest

└──▷ **BREAKING ON UPGRADE**

! First launch after upgrading triggers a one-time full reparse (parser version 13); the app is usable afterwards but the initial launch will be slower than usual.

WAS THIS USEFUL? 👍 👎

OpenAI Codex CLI

Feedback ⓘ rust-v0.162.0-alpha.13 NOTES ↗

CODE ↗ RANK

OpenAI Codex CLI runs an agent in the terminal that reads, changes, and tests code in local repositories.

> OpenAI Codex CLI's latest alpha build improves daemon control for `codex remote-control`, adds a Windows junction fallback for daemon release selection, and fixes reasoning-summary defaults propagation in embedded TUI threads.

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

01 Daemon control for codex remote-control NEW

85

Adds a `--no-daemon` flag to `codex remote-control` that explicitly runs the embedded app-server in the foreground until Ctrl-C instead of using the managed daemon. Plain `codex remote-control` (without a subcommand) now automatically starts or reuses the managed daemon when daemon auto-start is enabled and the environment is eligible, and the top-level `--no-daemon` flag is honored and propagated from the root CLI to the remote-control subcommand.

Run the remote-control app-server in the foreground (no background daemon) — useful in environments where detached daemon launch is restricted or for debugging.

```
$ codex remote-control --no-daemon
```

— Flag and subcommand named with a runnable example command rust-v0.162.0-alpha.13

02 Windows junction fallback for daemon release selection

IMPROVED

60

Falls back to `cmd.exe` with `mklink /J` on Windows when native junction creation or retargeting returns `PermissionDenied`, allowing managed daemon release selection to succeed under restrictive policies.

— Names mechanism and trigger condition but no example rust-v0.162.0-alpha.13

03 Reasoning summary defaults in embedded TUI IMPROVED

60

Correctly applies server reasoning summary defaults in new embedded TUI threads, forwarding explicit `model_reasoning_summary` and `features.concurrent_reasoning_summaries` choices from profile and command-line overrides.

— Names config fields but lacks usage example rust-v0.162.0-alpha.13

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

Feynman

Feedback ⓘ v0.5.26 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 two releases this window harden research reliability — paper search now survives Semantic Scholar/OpenAlex timeouts and fixes arXiv version resolution — while also shipping smarter Copilot model defaults, a `-p` prompt shorthand, an alphaXiv fallback for `alpha_ask_paper`, and web-fetching improvements via pi-web-access 0.36.0.

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

01 **Paper search survives index timeouts** IMPROVED 65

When a Semantic Scholar or OpenAlex topic search hits the 25 s timeout, Feynman now surfaces the unresponsive index name in research output instead of a generic abort message, and continues the research step using the other index.

— Explains before/after and names the 25s timeout and indices, no command. v0.5.25

02 **GitHub Copilot model default selection fixed** IMPROVED 65

GitHub Copilot logins now default to the newest available Claude Opus model, then Sonnet, then GPT, instead of falling back to alphabetical order and picking an unsupported model like `claude-haiku-4.5`.

Check your active Copilot model and switch away from an auto-selected `claude-haiku-4.5` to a supported model.

```
$ feynman status && feynman model set
```

— Clear before/after with named model fallback order and a check command. v0.5.26

03 **alphaXiv fallback for `alpha_ask_paper`** IMPROVED 60

`alpha_ask_paper` now answers with arXiv metadata and abstract as a fallback when alphaXiv is unavailable (e.g. after login expiry), matching the existing fallback behaviour of `alpha_search` and `alpha_get_paper`.

— Names three tools and the fallback mechanism, no usage example. v0.5.26

THINNER COVERAGE BELOW

04 **arXiv version resolution fix** IMPROVED 55

Reports unavailable explicit arXiv versions in `not_found` instead of silently resolving to a different version, and returns the latest version for a bare arXiv ID regardless of feed order.

— Names the `not_found` field and exact fix, but no example. v0.5.25

05 **Web search and fetch improvements (pi-web-access 0.36.0)**

IMPROVED 55

`web_search` now supports lists passed as a JSON string, `fetch_content` prefers markdown when requesting docs sites, and `fetch_content` now reports an error when its only URL fails — all shipped via the pi-web-access 0.36.0 update.

— Names three functions and the dependency version, no usage shown. v0.5.26

06 **`-p` shorthand for `--prompt`** IMPROVED 50

Adds `-p` as a short alias for `--prompt`, matching the style of Pi.

Pass a prompt inline from the shell without typing the full `--prompt` flag.

```
$ feynman -p 'what do we know about scaling laws'
```

— Trivial flag alias but comes with a runnable example command. v0.5.26

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Agentic Infrastructure

OpenRouter

Feedback ⓘ snapshot-20261005 NOTES ↗ RANK

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

> OpenRouter introduced server-side code execution tools that let any model run shell commands directly through OpenRouter's own sandbox, without the caller provisioning a container.

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

01 Server-side code execution tools for any model NEW

75

OpenRouter adds `openrouter:shell`, a server-side code execution tool for running model-issued commands in OpenRouter's sandbox, available to any model on both the Chat Completions and Messages APIs. It also adds `openrouter:bash`, the same kind of sandboxed command-execution tool but scoped to the Messages API. Both are invoked by passing a `tools` entry such as `{"type": "openrouter:shell"}` in the API request, letting a model list files or run commands without the caller provisioning or securing a container.

Use any model (e.g. a non-OpenAI/Anthropic model) with server-side code execution via the Chat Completions API — no container to provision or secure.

```
$ curl https://openrouter.ai/api/v1/chat/completions \
  -H 'Authorization: Bearer <your_api_key>' \
  -H 'Content-Type: application/json' \
  -d '{
    "model": "<any-model>",
    "tools": [{"type": "openrouter:shell"}],
    "messages": [{"role": "user", "content": "List files in the
current directory and return the output."}]
  }'
```

— Names both tool types and APIs but lacks sandbox limits/mechanism detail

snapshot-20261005

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.8.128 NOTES ↗ RANK

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

> HyperFrames shipped an importable font-pipeline module for custom builds, native HeyGen voice management, and deeper cross-platform Studio integration for render hand-off.

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

01 **Importable font pipeline module** NEW 75

Exposes the renderer's font pipeline as an importable module from `@hyperframes/core/fonts/embed`, letting consumers pull font-embedding logic directly into their own build.

— Named import path given but no usage example shown v0.8.128

THINNER COVERAGE BELOW

02 **HeyGen voice management via Media-use subsystem** NEW 55

Adds the ability to clone, list, and delete HeyGen voices and to accept a host OAuth token via the Media-use subsystem.

— Describes capability but no CLI or API call shown v0.8.124

03 **Cross-platform Studio hand-off and workflow integration** NEW 55

Enables CLI hand-off to Studio on macOS and opens the app on Windows and Linux, automatically opens Studio before every final render via the Skills workflow, and makes edits made with framey stand out in the Studio header.

— Several thin UI/workflow behaviors with no exact command syntax v0.8.124

WAS THIS USEFUL? 👍 👎

OpenAI TypeScript SDK

Feedback ⓘ v7.28.0 NOTES ↗

RANK

The OpenAI TypeScript SDK provides typed JavaScript and TypeScript clients for OpenAI APIs and streaming responses.

> OpenAI TypeScript SDK v7.28.0 expands the API surface with support for custom voice creation and agent session events.

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

01 **Custom voice creation support** NEW 20

Adds support for custom voice creation to the API surface.

— Only a bare mention, no endpoint or usage details v7.28.0

02 **Agent session events support** NEW 20

Adds support for agent session events to the API surface.

— Only a bare mention, no endpoint or usage details v7.28.0

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

vLLM

Feedback ⓘ v0.31.0 NOTES ↗ CODE ↗ RANK

The vLLM inference engine runs large language models with high-throughput, memory-efficient GPU execution.

> vLLM v0.31.0 adds a weight-cache daemon for fast engine restarts, new admission and all-to-all scheduling controls, expanded speculative decoding and large-scale MoE/parallelism infrastructure, broader LoRA coverage, and a run of breaking changes including mandatory opt-in for per-request multimodal kwargs, removal of slow tokenizer mode, and removal of several quantization backends.

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

01 Fast-restart weight-cache daemon via vllm preload NEW

80

Adds the `vllm preload` / `vllm snapshot` CLI subcommand to launch a weight-cache daemon that keeps post-quantized weights resident in GPU memory across engine restarts, with data-parallelism support, MTP draft model support, a `/health` endpoint, and a readiness wait.

— Names CLI subcommand and endpoint but no runnable example given v0.31.0

02 Removed and renamed CLI/config options BREAKING

80

`tokenizer_mode="slow"` has been removed; `--enable-mamba-fine-grained-prefix-cache` is renamed to `--enable-mamba-shared-prefix-checkpoint`; Quark silent online quantization has been removed; the AllSpark INT8 W8A16 backend has been removed; and `--enforce-eager` now also disables JIT kernel warmup, which may affect startup behavior for users who relied on JIT warmup under eager mode.

— Every removed/renamed flag and value named verbatim v0.31.0

03 Opt-in gating for per-request multimodal kwargs BREAKING

75

Adds `--trust-request-mm-kwargs`; per-request `mm_processor_kwargs` and `media_io_kwargs` are now rejected unless this flag is set, hardening the multimodal attack surface. This is a breaking change: existing deployments that pass per-request multimodal kwargs will fail without the flag.

— Exact flag and fields named, no migration command shown v0.31.0

04 **Model Runner V2 speculative decoding enhancements** NEW

75

Adds the LiLiCorr speculative-decoding drafter, draft-model speculative decoding and custom logits processors validated at admission, DFlash async scheduling with context K/V precompute captured in the draft CUDA graph, DSpark adaptive verification for Gemma4, variable-length decode for Kimi-K3, Cohere2MoE auxiliary hidden states for EAGLE3/DFlash drafters, and MiMo V2 MXFP4 MoE, BF16 MoE router, and DFlash draft support — all on Model Runner V2.

— Many named mechanisms but no command or config shown v0.31.0

05 **Active-sequence admission cap flag** NEW

70

Adds `--max-num-active-seqs` to cap RUNNING-queue admission independently of `max_num_seqs`, letting operators limit concurrently active sequences to avoid memory pressure from new admissions.

Cap the number of concurrently active sequences to prevent memory pressure from new admissions on a shared inference node.

```
$ vllm serve meta-llama/Llama-3-8B --max-num-active-seqs 64 --
max-num-seqs 256
```

— Runnable command shows exact flag usage v0.31.0

06 **MoonEP all-to-all backend for MoE** NEW

70

Adds `--all2all-backend moonep` to enable the MoonEP balanced EP all-to-all backend for large-scale MoE serving.

Route MoE all-to-all traffic through the MoonEP balanced backend for large expert-parallel deployments.

```
$ vllm serve deepseek-ai/DeepSeek-V4.1-Flash --all2all-backend
moonep --tensor-parallel-size 8
```

— Runnable command demonstrates the exact flag v0.31.0

07 **Prefix-cache key collision fix via LoRA-tagged hashing**

IMPROVED

60

Adds prefix-cache extra-key tagging by source so a LoRA name and a `cache_salt` can no longer collide; the LoRA path is now part of the block hash.

— Clear before/after but no config key to act on v0.31.0

08 **fp8_per_tensor online quantization shorthand** BREAKING

60

Adds the online quantization shorthand `fp8_per_tensor`, replacing the former `quantization="fp8"` online path; the old form no longer works.

— Exact config string change named, no full example v0.31.0

THINNER COVERAGE BELOW

09 **Large-scale MoE and parallelism infrastructure** NEW

55

Adds prefill context parallelism combined with data parallelism for large-scale serving, DeepEPv2 with sequence parallelism and EPLB with shared-expert overlap, a sharding-aware NCCL M2N weight-transfer backend for reinforcement-learning workloads, and KV offloading back-pressure detection.

— Named subsystems but no flags or commands given v0.31.0

10 **LoRA support expansion to new model types** NEW 55

Adds LoRA support for Nemotron VL language models, ModernBert, VoyageQwen3 embeddings, RoBERTa sequence classification, and `modules_to_save` sequence-classification heads with per-adapter `num_labels`.

— Lists several named model types, no config example v0.31.0

11 **Adaptive long-prefill chunking threshold** IMPROVED 50

The `--long-prefill-token-threshold` behavior now has the scheduler adapt the threshold to the number of waiting prefills instead of always chunking a lone request.

— Flag named, but no concrete usage example v0.31.0

12 **FIM completion via suffix field for DeepSeek V4** NEW

50

Adds fill-in-the-middle completion support via the `suffix` field for DeepSeek V4.

— Names the field but no request example shown v0.31.0

13 **XPU CUDA graphs enabled by default** BREAKING 50

Enables XPU CUDA graphs by default, previously opt-in via the `VLLM_XPU_ENABLE_XPU_GRAPH` environment variable, which has now been removed.

— Names removed env var but no migration step v0.31.0

14 **Granite thinking reasoning parser** NEW 40

Adds the built-in `granite_thinking_parser` reasoning parser for Granite 4.2.

— Names the parser but not the flag to select it v0.31.0

15 **KV-aware waiting queue reordering** IMPROVED 40

Reworks the waiting queue so requests already holding KV blocks are scheduled first, reducing recomputation under memory pressure.

— Clear before/after but no named surface to use v0.31.0

16 **DiffusionGemma structured generation mode** NEW 35

Adds a DiffusionGemma structured generation mode with bounded single-token choices.

— Brief description, no usage detail given v0.31.0

17 **Encoder CUDA graphs for vision tower** NEW 35

Adds encoder CUDA graphs for the vision tower on DeepSeek-V4.1-Flash and other multimodal models.

— Names affected models but no flag or config shown v0.31.0

└──▷ **BREAKING ON UPGRADE**

! Per-request `mm_processor_kwargs` and `media_io_kwargs` are now rejected unless `--trust-request-mm-kwargs` is set; existing deployments that pass per-request multimodal kwargs will break without this flag.

! `tokenizer_mode="slow"` has been removed; any configuration using this value will fail.

! `--enable-mamba-fine-grained-prefix-cache` is renamed to `--enable-mamba-shared-prefix-checkpoint`; scripts using the old flag name will break.

! Online quantization via `quantization="fp8"` is replaced by the `fp8_per_tensor` shorthand; the old form no longer works.

! Quark silent online quantization has been removed.

! The AllSpark INT8 W8A16 backend has been removed.

! `--enforce-eager` now also disables JIT kernel warmup, which may affect startup behavior for users who relied on JIT warmup under eager mode.

! `VLLM_XPU_ENABLE_XPU_GRAPH` environment variable has been removed; XPU graphs are now always enabled by default on XPU.

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

okf-memory OKF Agent Memory

Feedback ⓘ v0.6.0

NOTES ↗ RANK

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

> OKF Agent Memory's v0.6.0 release adds membership-based list filtering and total-file token reporting to its CLI, while tightening several security boundaries around symlinked roots, human verification claims, and output schemas.

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

01 **List-field membership matching in --filter** NEW 87

`--filter` now supports membership matching against list-valued custom frontmatter fields: `topics=retrieval` matches when the list contains the value (case-insensitive), `topics!=retrieval` matches when it does not, and `topics=null` matches an empty list; repeating the key ANDs multiple values, e.g. `--filter "tags=ci,tags=ui"`.

Filter concepts by membership in a list-valued custom field — useful when tagging concepts with topics and needing to scope a search to a specific one.

```
$ ./bin/okf search "retrieval" knowledge --filter
"topics=retrieval"
```

Require multiple tags simultaneously in a filter to narrow results to concepts that satisfy all of them.

```
$ ./bin/okf search "auth" knowledge --filter
"tags=ci,tags=security"
```

— Full matching semantics with runnable filter examples. v0.6.0

02 **Total file token reporting in lint output** NEW 78

`okf agents lint` and `okf validate --agents` now report `token_stats.total_tokens` in `--json` output alongside managed-block tokens, and show `M in total file` next to the managed-block count in plain-text output, surfacing content outside the `<!-- BEGIN OKF AGENT MEMORY -->` / `<!-- END OKF AGENT MEMORY -->` markers.

Audit total file token load (not just the managed block) in CI to catch bloat outside the `<!-- BEGIN OKF AGENT MEMORY -->` block.

```
$ ./bin/okf agents lint --json | jq '.token_stats.total_tokens'
```

— Named JSON field plus a runnable jq audit command. v0.6.0

03 **Human verification enforcement tightened** BREAKING 70

`SaveConcept` now rejects `verified` entries naming a human (`human`, `human:*`, `human/*`, any case) when the writing actor is not human, and the MCP server actor is fixed to `agent/mcp`. Correspondingly, `--filter verified.by=human` no longer matches near-miss values such as `humanoid-bot`; only `human`, `human:*`, and `human/*` (any case) qualify.

— Mechanism and matching values named, no runnable example. v0.6.0

04 **Custom list fields now typed JSON arrays** BREAKING 67

Custom list frontmatter fields such as `extra.commands` and `extra.repos` are now returned as typed JSON arrays in `okf show --json` output and MCP tools, replacing the prior format of strings with `- ` prefixes; this changes the JSON schema for any existing consumer of those fields.

— Before/after schema named, no migration command given. v0.6.0

05 **Undocumented --filter aliases removed** BREAKING 65

The undocumented `--filter` aliases `tag` (for `tags`), `code_ref` (for `code_refs`), and `desc` (for `description`) are removed; callers must use `tags=`, `code_refs=`, and `description=` instead. An old `--filter "tag=security"` invocation now silently matches nothing rather than erroring.

— Exact aliases and replacements named, no runnable example. v0.6.0

06 **Symlinked project roots must stay in-directory** BREAKING 65

A symlinked project or vendor bundle root must now stay inside the directory containing the link; a link such as `knowledge -> /somewhere/else` is rejected. User/system scope roots (`~/.okf`, `/etc/okf`, `OKF_USER_DIR`, `OKF_SYSTEM_DIR`) remain exempt and may still symlink to any location.

— Exact paths and env vars named, no example shown. v0.6.0

└──▷ BREAKING ON UPGRADE

! The undocumented `--filter` aliases `tag` (for `tags`), `code_ref` (for `code_refs`), and `desc` (for `description`) are removed; use `tags=`, `code_refs=`, and `description=` instead. An old `--filter "tag=security"` now silently matches nothing.

! A symlinked project or vendor bundle root must stay inside the directory that contains the link; `knowledge -> /somewhere/else` is now rejected. User/system scope roots (`~/.okf`, `/etc/okf`, `OKF_USER_DIR`, `OKF_SYSTEM_DIR`) may still symlink to any location.

! `SaveConcept` now rejects `verified` entries naming a human (`human`, `human:*`, `human/*`, any case) when the actor is not human; affects library callers and the CLI when an agent actor writes human verifications. The MCP server actor is fixed to `agent/mcp`.

! `--filter verified.by=human` no longer matches values such as `humanoid-bot`; only `human`, `human:*`, and `human/*` (any case) qualify as human.

! `okf show --json` and MCP tools now return custom block-list fields as typed JSON arrays instead of strings with `- ` prefixes, which changes the JSON schema for any consumer of those fields.

WAS THIS USEFUL? 👍 👎

GrowGraph OntoCast

Feedback ⓘ v0.6.7 CODE ↗ RANK

OntoCast generates RDF knowledge graphs from documents by co-evolving domain ontologies and fact graphs with an agentic LLM pipeline.

> OntoCast's documentation window surfaces a large configuration layer — covering render-mode selection, concurrency and rate-limiting, ontology text budgeting, document conversion, critic passes, storage and triple-store backends — alongside two shipped releases: a disk-backed LLM response cache with CLI management, and a breaking change switching the default LLM graph output from JSON-LD to compact Turtle.

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

01 Default LLM graph output switches to compact Turtle

BREAKING

91

Adds `LLM_GRAPH_FORMAT` config key (`turtle` | `jsonld` , default `turtle`) controlling the RDF serialisation the LLM writes, and `LLM_OUTPUT_LAYOUT` (`compact` | `free` , default `compact`) controlling whitespace to reduce billed output tokens; `compact` requests minified JSON and one-line-per-subject Turtle, recorded in the run manifest's `prompting` block. v0.6.7 changes the default from `LLM_GRAPH_FORMAT=jsonld` (no layout) to `LLM_GRAPH_FORMAT=turtle` with `LLM_OUTPUT_LAYOUT=compact` , invalidating cached LLM responses; set `LLM_GRAPH_FORMAT=jsonld` and `LLM_OUTPUT_LAYOUT=free` to restore the previous behaviour. Adds `rdf/turtle_repair` , `rdf/jsonld_rdflib_fallback` , and `repair/compact_iri_literal` counters for payloads that needed recovery before parsing.

Switch to JSON-LD graph output and compact layout to minimize token spend on providers whose structured-output handles nested objects better than long Turtle strings.

```ini
LLM_GRAPH_FORMAT=jsonld
LLM_OUTPUT_LAYOUT=compact
ONTOLOGY_CHAPTER_FORMAT=term_sheet
RENDER_MODE=facts
```

Restore pre-v0.6.7 JSON-LD output if your pipeline depends on the old format and you need to reuse cached LLM responses.

📍 `LLM_GRAPH_FORMAT=jsonld` `LLM_OUTPUT_LAYOUT=free` python `-m`

ontocast …

Explicitly opt in to compact Turtle output (the new default) to minimise token usage in LLM responses.

📍 `LLM_GRAPH_FORMAT=turtle` `LLM_OUTPUT_LAYOUT=compact` python `-m` ontocast …

— Full before/after default change with exact restore command given `v0.6.7`

---

02 **Disk-backed LLM response cache with CLI management** NEW

91

Adds disk-backed LLM response caching under `llm/`, `converter/`, and `chunker/` subdirectories, keyed on the full prompt text and relevant LLM settings (`LLM_PROVIDER`, `LLM_MODEL_NAME`, `LLM_TEMPERATURE`, `LLM_BASE_URL`, `LLM_THINK`, `LLM_NUM_PREDICT`, `LLM_NUM_CTX`, `LLM_REASONING_EFFORT`, `LLM_THINKING_BUDGET`, `LLM_JSON_MODE`, and output schema name), so identical requests are answered from disk at zero token cost. Configurable via `ONTOCAST_CACHE_MAX_BYTES`, `ONTOCAST_CACHE_TTL_DAYS`, and `ONTOCAST_CACHE_DIR`, with `LLM_CACHE_READ_ONLY=true` locking the cache for reproducible runs, and managed with the `ontocast cache stats` and `ontocast cache prune --orphaned --ttl-days 14` CLI commands.

Cap the cache at 500 MB and expire entries unused for 30 days so a long-running server stays bounded without manual cleanup.

```
$ ONTOCAST_CACHE_MAX_BYTES=500MB ONTOCAST_CACHE_TTL_DAYS=30
ontocast serve
```

Redirect the cache to a shared volume for a run, then inspect per-subdirectory usage and trim stale orphaned entries.

```
$ ONTOCAST_CACHE_DIR=/data/ontocast-cache ontocast serve
# later, in another shell:
ontocast cache stats
ontocast cache prune --orphaned --ttl-days 14
```

Lock the cache for a reproducible benchmark — reads from a pre-warmed cache but never writes new entries, so results are stable across repeated runs.

```
$ LLM_CACHE_READ_ONLY=true ontocast serve
```

— Named keying mechanism plus runnable CLI commands product docs

03 **Concurrency and rate-limit controls** NEW 75

Adds `MAX_CONCURRENT_PROCESSES` to cap simultaneous `/process` and `/process_unit` server requests, `PARALLEL_WORKERS` to set how many content units of one document process concurrently, `LLM_MAX_INFLIGHT` to cap provider calls in flight across the whole process, and `LLM_MAX_RETRIES` to configure the provider SDK retry budget with back-off and Retry-After support on rate limits and dropped connections.

Throttle provider calls and pace requests when hitting a rate-limited tier, so the SDK retries handle transient 429s without hammering the endpoint.

```
$ export LLM_MAX_INFLIGHT=4
export LLM_REQUESTS_PER_SECOND=8
export LLM_MAX_RETRIES=5
export LLM_REQUEST_TIMEOUT_SECONDS=120
```

Cap ontology text literals to control prompt size for large catalogs, and limit server concurrency to avoid hitting provider rate limits.

ini

```
ONTOLOGY_TEXT_MAX_CHARS_NAMING=80
ONTOLOGY_TEXT_MAX_CHARS_CONTRACT=200
ONTOLOGY_TEXT_MAX_CHARS_PROSE=150
ONTOLOGY_TEXT_TOTAL_BUDGET=4000
MAX_CONCURRENT_PROCESSES=4
PARALLEL_WORKERS=8
```

— Four env vars named with a runnable throttling example product docs

04 **Render mode selects ontology-only, facts-only, or both** NEW 67

Adds `RENDER_MODE` config key (values `ontology`, `facts`, `ontology_and_facts`; default `ontology_and_facts`) to select what the LLM renders per content unit, letting a run extract facts against an existing ontology without re-rendering it, or vice versa.

Run only fact extraction against your existing catalog (no ontology changes) with 8 parallel workers and a tighter chunk size.

```
$ RENDER_MODE=facts PARALLEL_WORKERS=8 CHUNK_MIN_SIZE=500
CHUNK_MAX_SIZE=2000 ontocast process <path>
```

— Exact values and default with a runnable example command product docs

05 **Document conversion and chunking controls** NEW 64

Exposes `CHUNK_MIN_SIZE` and `CHUNK_MAX_SIZE` to control the character bounds of content units fed to each LLM prompt, `CONVERTER_PROFILE=auto` (with a `lean` alternative) to control PDF OCR behavior and equation handling during the Convert step, `CONVERTER_DO_FORMULA_ENRICHMENT` to decode equations to LaTeX, and `CONVERTER_SUPPORTED_EXTENSIONS` to declare which file suffixes are routed through the document converter.

Run only fact extraction against your existing catalog (no ontology changes) with 8 parallel workers and a tighter chunk size.

```
$ RENDER_MODE=facts PARALLEL_WORKERS=8 CHUNK_MIN_SIZE=500
CHUNK_MAX_SIZE=2000 ontocast process <path>
```

Enable equation-to-LaTeX enrichment and post-processing critic passes when ingesting scientific PDFs to maximize facts recall.

```
$ CONVERTER_PROFILE=auto CONVERTER_DO_FORMULA_ENRICHMENT=true
FACTS_CRITIC_PASSES=2 FACTS_COMPLETION_PASSES=1 ontocast process
paper.pdf
```

— Five keys named with runnable PDF-processing example product docs

06 **Ontology text length budgeting for prompts** NEW 62

Adds `ONTOLOGY_TEXT_MAX_CHARS_NAMING`, `ONTOLOGY_TEXT_MAX_CHARS_CONTRACT`, `ONTOLOGY_TEXT_MAX_CHARS_PROSE`, and `ONTOLOGY_TEXT_TOTAL_BUDGET` config keys to bound `skos:scopeNote` and `skos:definition` text length and total ontology text injected per prompt, controlling prompt size for large catalogs.

Reduce prompt token cost on a catalog with long definitions by capping ontology text fields and switching to a term-sheet chapter format.

```
$ export ONTOLOGY_CHAPTER_FORMAT=term_sheet
export ONTOLOGY_TEXT_MAX_CHARS_NAMING=80
export ONTOLOGY_TEXT_MAX_CHARS_CONTRACT=200
export ONTOLOGY_TEXT_MAX_CHARS_PROSE=120
export ONTOLOGY_TEXT_TOTAL_BUDGET=500
```

Cap ontology text literals to control prompt size for large catalogs, and limit server concurrency to avoid hitting provider rate limits.

```ini
ONTOLOGY_TEXT_MAX_CHARS_NAMING=80
ONTOLOGY_TEXT_MAX_CHARS_CONTRACT=200
ONTOLOGY_TEXT_MAX_CHARS_PROSE=150
ONTOLOGY_TEXT_TOTAL_BUDGET=4000
MAX_CONCURRENT_PROCESSES=4
PARALLEL_WORKERS=8
```

— Four keys named with a runnable example setting all four `product docs`

07 **Run telemetry counters in /process responses and manifests**

NEW 60

Exposes `llm/rate_limited`, `llm/timeouts`, `llm/ontology_selection`, `llm/parse_retry`, `prefix_cache_hit_rate`, `chapter/text_chars_before`, `chapter/text_chars_after`, `loop_lag_total`, and `reasoning_share_of_output` telemetry counters in the `/process` response and run manifest for diagnosing performance and cost; `POST /process` returns this telemetry alongside the ontologies and facts graphs, and the `ontocast process` CLI command writes the run manifest to disk.

— Names ten counters and two surfaces but no consumption example `product docs`

THINNER COVERAGE BELOW

08 **Ontology context retrieval mode for vector-backed prompting**

NEW 57

Adds `ONTOLOGY_CONTEXT_MODE` (e.g. `selected_vector_search_ontology`), `ONTOLOGY_CONTEXT_SCOPE`, `FANOUT_WARMUP_UNITS`, `ONTOLOGY_CONTEXT_MAX_TRIPLES`, and `ONTOLOGY_CONTEXT_REQUIRED` config keys to select vector-store ontology context scoped to a document, with warm-up priming so provider prefix caches serve all but the first parallel call.

Use vector-store ontology context with document-scoped chapters and warm-up priming so provider prefix caches serve all but the first parallel call.

```ini
ONTOLOGY_CONTEXT_MODE=selected_vector_search_ontology
ONTOLOGY_CONTEXT_SCOPE=document
FANOUT_WARMUP_UNITS=2
ONTOLOGY_CONTEXT_MAX_TRIPLES=500
ONTOLOGY_CONTEXT_REQUIRED=true
```

— Five keys named but only shown via one example, no bullet description product docs

09 **Critic and completion passes for facts and ontology quality** NEW 53

Exposes `FACTS_CRITIC_PASSES` and `ONTOLOGY_CRITIC_PASSES` to configure how many critic-review iterations run per block (facts on by default, ontology off), and `FACTS_COMPLETION_PASSES` to enable additive completion passes that recover measurements missed in the initial facts render.

Enable equation-to-LaTeX enrichment and post-processing critic passes when ingesting scientific PDFs to maximize facts recall.

```
$ CONVERTER_PROFILE=auto CONVERTER_DO_FORMULA_ENRICHMENT=true
FACTS_CRITIC_PASSES=2 FACTS_COMPLETION_PASSES=1 ontocast process
paper.pdf
```

— Three keys named with defaults, one example invokes them product docs

10 **Shared embedding model reduces peak memory** IMPROVED 53

Documents that setting the same model name in `CHUNK_EMBEDDING_MODEL`, `EMBEDDING_MODEL_NAME`, and `AGG_EMBEDDING_MODEL` reduces resident models from two to one, lowering peak memory; `EMBEDDING_PROVIDER=ollama` can offload retrieval entirely if memory is still tight.

Keep peak memory low on a machine running all three embedding uses by pointing them all at one shared model, and offload retrieval to Ollama if memory is still tight.

```
$ export CHUNK_EMBEDDING_MODEL=sentence-transformers/paraphrase-
multilingual-MiniLM-L12-v2
export EMBEDDING_MODEL_NAME=sentence-transformers/paraphrase-
multilingual-MiniLM-L12-v2
export AGG_EMBEDDING_MODEL=sentence-transformers/paraphrase-
multilingual-MiniLM-L12-v2
# or offload retrieval entirely:
# export EMBEDDING_PROVIDER=ollama
```

— Mechanism and before/after memory effect explained with example product docs

11 **Expanded input format support** NEW 45

Supports PDF, Word, PowerPoint, Excel, HTML, Markdown, CSV, AsciiDoc, and image input formats via the `doc-processing` extra, plus native `.txt`, `.json`, and `.jsonl` reading without it.

– Lists every format and the extra name, no command shown product docs

12 **Apache Jena Fuseki triple store backend support** NEW

40

Supports Apache Jena Fuseki as an external triple store backend in addition to in-memory storage, with output serialized as Turtle.

– Names the backend but no config key for selecting it product docs

13 **Retry limit for failed content-unit renders** NEW 37

Adds `MAX_VISITS_PER_NODE` (also read from `MAX_VISITS`) config key to set the retry count for renders that fail outright — an unparseable response or a provider error — on any single content unit.

– Names the key and its fallback alias, no usage example product docs

14 **Per-document named-graph storage with ontology versioning**

NEW 35

Stores facts in a per-document named graph and ontologies as versioned snapshots, with provenance tracking each content unit's contribution to the graph separately.

– Describes storage mechanism but no config keys or commands product docs

└──▷ **BREAKING ON UPGRADE**

! The default output encoding changes from `LLM_GRAPH_FORMAT=jsonld` (no layout instruction) to `LLM_GRAPH_FORMAT=turtle` with `LLM_OUTPUT_LAYOUT=compact`; cached LLM responses are not reused. Set `LLM_GRAPH_FORMAT=jsonld` and `LLM_OUTPUT_LAYOUT=free` to restore the previous behaviour.

WAS THIS USEFUL?  

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ snapshot-20261005 NOTES ↗

RANK

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

> LangSmith added a new `/v2/threads/stats` API endpoint for pulling per-project latency, token and cost metrics, extended LLM Gateway with `ls_provider` tagging and a standard `ChatOpenAI` manifest for Gateway-backed prompts, and improved dashboards with multi-source chart pooling and custom dashboard linking, while tightening dataset export permissions.

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

01 **/v2/threads/stats API endpoint** NEW 78

A new `GET /v2/threads/stats?project_id=<project_id>` endpoint returns latency, token, and cost statistics for a tracing project, letting teams pull the data into monitoring dashboards or alert pipelines.

Pull latency, token, and cost statistics for a tracing project to feed into a monitoring dashboard or alert pipeline.

```
$ curl -X GET 'https://api.smith.langchain.com/v2/threads/stats?
project_id=<project_id>' \
  -H 'x-api-key: <LANGSMITH_API_KEY>'
```

— exact endpoint and runnable curl example given snapshot-20261005

02 **Standard ChatOpenAI manifest for Gateway prompts** IMPROVED 65

Prompts saved with a LangSmith Gateway model now use a standard `ChatOpenAI` manifest with the unified Gateway URL and a `LANGSMITH_GATEWAY_API_KEY` secret reference, making them compatible with standard LangChain SDKs.

— named manifest and secret key, no usage example snapshot-20261005

03 **Dashboard chart data pooling across sources** NEW 60

LangSmith dashboards can now pool runs from multiple tracing projects or agent environments into a single chart data set instead of displaying each source separately, and agent-based workspaces gain a `+ Select environment or dataset` button in the chart data popover to switch between tracing and dataset sources.

— names UI control but no exact navigation path `product docs`

04 **Custom dashboard linking from tracing projects** NEW 60

Any custom dashboard can now be linked directly from a tracing project or agent environment, overriding the default prebuilt dashboard; click the three dots next to the dashboard name and select the custom dashboard to link it.

Link a custom dashboard to an agent environment instead of the default prebuilt one.

> 📍 In your tracing project or agent environment, click the three dots next to the dashboard name, then select your custom dashboard to link it.

— clear step-by-step UI instructions provided `product docs`

THINNER COVERAGE BELOW

05 **Resource tag assignment for Application tag** NEW 45

Workspace editors can now assign resources to an Application tag value from Settings > Resource Tags.

— UI path given but no further mechanism `snapshot-20261005`

06 **IAM-role based ECR registries** NEW 20

LangSmith adds support for IAM-role based ECR registries.

— named only in summary, no mechanism detailed `snapshot-20261005`

07 **Gateway fallback timeout controls** NEW 20

LangSmith adds Gateway fallback timeout controls.

— named only in summary, no detail given `snapshot-20261005`

└──▷ **BREAKING ON UPGRADE**

! Dataset CSV and JSONL exports now require both dataset read and download permissions — API requests that previously succeeded with only dataset read permission now return a forbidden response.

WAS THIS USEFUL? 👍 👎

eunomia-bpf AgentSight

Feedback ⓘ v1.0.33 NOTES ↗

RANK

lightweight system-level observability for AI Agents

> AgentSight added CodeBuddy CLI integration with TLS and file-system visibility, extended its eBPF process probe with network-aware socket capture and resolved `FILE_OPEN` tracing, and added dual-format SSL payload fields to sslsniff.

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

01 Network-aware process capture via --trace-net IMPROVED

95

A `--trace-net` flag was added to `sudo ./agentsight debug trace --trace-net --server` and `sudo ./bpf/process --trace-net`, capturing network summaries including attempted bind/connect, successful TCP listener local endpoints, and accepted peer endpoints tagged with `pid` and `process_start_ns`. The process probe extends this to summarise TCP listeners, accepted peers, and successful UDP or ICMP echo binds, with datagram `NET_BIND` records gaining `protocol` and kernel-assigned `port` fields; a later release further extends the eBPF process probe to capture additional inbound socket evidence for richer network context.

Enable network-aware process capture to see bind/connect attempts and TCP listener endpoints alongside process events.

```
$ sudo ./agentsight debug trace --trace-net --server
```

Enable network summarisation to see TCP listeners, accepted peers, and UDP binds alongside process events, then filter for `NET_BIND` records to inspect kernel-assigned ports.

```
$ agentsight process --trace-net | jq 'select(.event ==
"NET_BIND") | {protocol, port}'
```

— Exact flags, commands and JSON field names given v1.0.33

02 CodeBuddy CLI integration and session capture NEW

92

AgentSight now supports Tencent CodeBuddy CLI (`@tencent-ai/codebuddy-code`), reading session transcripts from `~/.codebuddy/projects/<project>/<session-`

`id>.jsonl` with a home directory override via the `CODEBUDDY_CONFIG_DIR` environment variable. It can attach directly to a running CodeBuddy session with `sudo ./agentsight record -c codebuddy`, targeting the CLI's Node process, and TLS capture surfaces `POST /chat/completions` traffic to `codebuddy.ai` / `copilot.tencent.com`.

Override the default CodeBuddy config home to point AgentSight at session transcripts stored in a non-standard location.

```
$ CODEBUDDY_CONFIG_DIR=/mnt/devbox/.codebuddy agentsight top
```

Capture all file and process activity from a running CodeBuddy CLI session to audit what paths the agent reads or executes.

```
$ sudo ./agentsight record -c codebuddy
```

— Names config path, env var, endpoint and exact attach command product docs

---

03 **FILE_OPEN tracing with resolved path enrichment** IMPROVED 87

The process probe adds `FILE_OPEN` JSON records with fields `read`, `write`, `exec`, `layer`, `dev`, and `ino` on supported kernels, covering `open`, `openat2`, io_uring, and exec including the ELF interpreter, plus `FILE_OPEN_LOST` records counting ring-buffer drops and BPF recursion misses. These events are further enriched with resolved file paths, read/write/exec access flags, and overlay layer/inode metadata.

Capture file-open events across a running workload to audit which files each process reads, writes, or executes — including ELF interpreters and io_uring paths.

```
$ agentsight process | jq 'select(.event == "FILE_OPEN")'
```

— Names every JSON field and syscall path covered product docs

---

04 **Dual SSL payload fields in sslsniff** NEW 65

sslsniff now emits two fields for every non-empty captured SSL read or write: `data`, a readable JSON text representation, and `datahex`, the exact captured plaintext as lowercase hexadecimal.

Inspect raw SSL payload bytes captured by sslsniff — use datahex to recover the exact plaintext byte sequence when the data field contains non-printable content.

```
$ agentsight sslsniff | jq 'select(.datahex != null) | {data, datahex}'
```

— Names both fields but limited mechanism detail product docs

---

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ snapshot-20261005 NOTES ↗

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

Adds Coding Agent Insights dashboard, SQL charts builder, Loop-on-dashboards, and DPAPI credential storage for Windows bt CLI v0.22.2.

Adds a built-in Coding Agent Insights dashboard (public preview, requires data plane v2.13.0+) showing usage by user, API key, model, tool, and branch with cost breakdowns.

NEW INTEGRATION Adds a daily Loop automation for Coding Agent Insights that surfaces recurring problems and inefficiencies across team sessions.

Adds SQL charts (public preview, Pro/Enterprise only, requires data plane v2.1.1+) to build dashboard charts from a SQL query with a visual builder, plotting results as time series, top lists, or tables.

Adds Loop-on-dashboards so you can open Loop from an individual dashboard or chart editor to investigate data and build or refine charts in natural language.

Adds `%USERPROFILE%\.braintrust\auth\credentials.dpapi` DPAPI-encrypted credential storage on Windows so `bt` CLI profiles are shared between terminal sessions and packaged (MSIX) desktop apps.

+8 more

NEW INTEGRATION Claude Agent SDK verbatim_prompts tracing

NEW INTEGRATION AgentScope TeamPipeline.reply_stream() tracing

UPDATED INTEGRATION Claude Code subagent tracing nesting

UPDATED INTEGRATION GPT-6.1 Sol model availability in Loop and scheduled automations

└─▷ **BREAKING ON UPGRADE**

! After downgrading `bt` from v0.22.2 to an earlier version on Windows, saved credentials are lost and you must log in again.

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

ai-safe2-framework

Feedback ⓘ nexus-v0.5.0 NOTES ↗ RANK

Ai-Safe2-Framework provides governance, risk, and compliance controls for securing agentic AI systems and non-human identities.

> NEXUS v0.5.0 introduces the Sovereign Payment Gateway, a deterministic, fail-closed security boundary composed of ten control components that govern AI agent payment execution, alongside a machine-readable documentation manifest for automated agent discovery.

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

01 Sovereign Payment Gateway for AI agent payments NEW

86

Adds the `nexus-a2a-sdk==0.5.0` Python package (`import nexus_sdk`) implementing a monotonic, fail-closed transaction lifecycle for AI agents initiating or influencing payments. It composes the Gateway State Vault (durable tracking of authority, exposure, replay, revocation, reservation and settlement state), Key Guardian (isolates signing authority from the agent, signing only approved canonical transactions), Policy Authority (independent deterministic policy evaluation), Runtime Verifier (confirms the executing workload matches an approved, sufficiently fresh measurement before value moves), Settlement Truth Authority (obtains payment outcomes from authoritative external sources rather than trusting agent-reported results), Policy Parity Guard (detects divergence between policy implementations before activation), Sovereign Payment Coordinator (composes all controls into an ordered, fail-closed execution path), Governed Recovery (handles ambiguous settlement without unsafe retries or premature restoration of spending capacity), Sandbox Rail Readiness (prevents unverified payment integrations from being promoted into active use), and Deployment Proof and Readiness (produces a signed, independently verifiable activation decision before production deployment). It is described as covering threats including stolen credentials, compromised runtimes, replay and duplicate-payment attacks, authority escalation by child agents, spending split across agents to evade limits, protocol and checkout downgrades, and revocation races.

— Names package, import and ten mechanisms, but no runnable example nexus-v0.5.0

THINNER COVERAGE BELOW

02 **Machine-readable documentation manifest** NEW 48

Adds an `llms.txt` retrieval index and versioned machine-readable documentation manifest under the `/NEXUS` canonical directory to support automated agent discovery of framework documentation.

— Names file and path but no usage detail beyond that `nexus-v0.5.0`

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Open-Weight Models

DeepSeek V4 Flash 0731 Feedback ⓘ

context change · Oct 4, 2026 MODEL ↗ STABLE

DeepSeek V4 Flash 0731 is a chat model from DeepSeek.

Context window shrank from 1311K to 1049K tokens on OpenRouter (deepseek/deepseek-v4-flash-0731), first seen 4 Oct 2026.

▾ 1 more release in this issue · 2026-10-04

> price increase · Oct 4, 2026 MODEL ↗ STABLE
>
> Price increase on OpenRouter (deepseek/deepseek-v4-flash-0731): $0.021 → $0.0152 in, $0.32 → $1.28 out per million tokens (+300%), first seen 4 Oct 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3.8 27B

Feedback ⓘ price cut · Oct 4, 2026

MODEL ↗

Qwen3.8 27B is a multimodal model from Alibaba.

Price cut on OpenRouter (qwen/qwen3.8-27b): $0.42 → $0.425 in, $3.00 → $2.55 out per million tokens (-15%), first seen 4 Oct 2026.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · October 5, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe | Feedback on this issue