THE AI TOOLCHAIN NO. 043

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED OCTOBER 1, 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 216 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 40 of 42 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. Cline now pages oversized MCP tool output through an in-memory cache behind `cline://cache/` URIs ad of blowing up the context window. mlx-serve runs GGUF weights

vim ? help

on the native MLX engine with up to 66% faster decode. And Unsloth's packed INT4/MXFP4 path trains gpt-oss-120b on a single B200.

DEPLOY

Run GGUF checkpoints on Apple silicon and fine-tune a 120B model on one GPU

mlx-serve loads GGUF directly on the native MLX engine — no reconversion to MLX format before serving locally — and reports up to 66% faster decode and 51% faster prefill. Unsloth's packed INT4/MXFP4 training keeps weights quantized through the training loop, so gpt-oss-120b fits a single B200 instead of a multi-GPU node.

mlx-serve · Unsloth

BUILD

Keep a giant MCP tool response from eating the agent's context

A single verbose MCP tool call can consume the whole window and derail a run; Cline now stores the full result in an in-memory cache, hands the model a paged preview, and exposes the rest via `cline://cache/` URIs for retrieval on demand. Agno flips MCP tool exposure to opt-in, so an added server no longer silently injects dozens of tool definitions into every prompt.

Cline · Agno

GOVERN

Give each MCP server its own credentials and its own blast radius

Pi adds provider-based bearer-token auth and Anthropic workload identity federation, Copilot CLI scopes MCP authentication plus session-level directory read approvals, and gram targets risk policies and guardrails at individual MCP toolsets. The shared win: one agent talking to five servers stops sharing one token and one filesystem scope across all of them.

Pi · GitHub Copilot CLI

OFFENSIVE SECURITY

◆ Exploitation & C2

xalgorix

Feedback ⓘ v4.6.128 NOTES ↗ RANK

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

> xalgorix v4.6.128 adds a one-click settings export/import flow for migrating a full configuration — environment variables, LLM auth profiles, and key-store entries — between installs.

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

01 **Settings export/import for install migration** NEW 72

A new Backup & migration card under Settings → Environment exports the complete configuration as a JSON file, including all set environment variables (secrets in cleartext, readable only by the operator session; machine API tokens are blocked), all LLM auth profiles (including OAuth tokens), and multi-provider key-store entries. The matching Import flow merges a previously exported JSON bundle into a fresh install: unknown keys and masked values are skipped with a per-section report, environment application is atomic, and values absent from the file are left untouched.

Migrate a full xalgorix configuration — env vars, LLM auth profiles, and key-store entries — to a fresh install without re-entering credentials.

📍 In the source install: Settings → Environment → Backup & migration card → Export. Transfer the downloaded JSON file to the new host. In the destination install: Settings → Environment → Backup & migration card → Import, then select the JSON file.

— Names exact UI path and import/export behaviour, no API or flag names. v4.6.128

v4.6.128 NOTES ↗ LATEST STABLE

xalgorix v4.6.128 adds one-click settings export/import for install migration via a new UI card.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.6.128
https://github.com/xalgorix/xalgorix.git
# already have the repo? check out this version:
$ git checkout v4.6.128
```

└──▷ HOW TO FIND IT

Migrate a full xalgorix configuration — env vars, LLM auth profiles, and key-store entries — to a fresh install without re-entering credentials.

> 📍 In the source install: Settings → Environment → `Backup & migration` card → Export. Transfer the downloaded JSON file to the new host. In the destination install: Settings → Environment → `Backup & migration` card → Import, then select the JSON file.

Adds a `Backup & migration` card under Settings → Environment that exports the complete configuration as a JSON file — including all set environment variables (secrets in cleartext, readable only by the operator session; machine API tokens are blocked), all LLM auth profiles (including OAuth tokens), and multi-provider key-store entries.

Adds an Import flow that merges a previously exported JSON bundle into a fresh install: unknown keys and masked values are skipped with a per-section report, env application is atomic, and values absent from the file are never touched.

WAS THIS USEFUL? 👍 👎

BUILD

◆ AI Coding Agents

yetone magpie

Feedback ⓘ v0.1.595 CODE ↗ LATEST STABLE

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 v0.1.595 adds IPv4 fallback for broken IPv6 hosts and a new `own` provider-suffix mode for model names.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.595 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.595
```

Adds `SuffixOwn` (own) as a third Provider in model names mode — user-given custom model names display exactly as written, while vendor models still append their provider suffix.

Adds automatic IPv4 fallback for hosts where IPv6 connections are accepted but dropped before any data returns, keeping the fallback active for 30 minutes per host:port.

NEW INTEGRATION Muse Code (Meta muse CLI) agent integration

▾ 24 more releases in this issue · 2026-09-30 → 2026-10-01

v0.1.593 CODE ↗ STABLE

magpie v0.1.593 adds a Sessions page for browsing, resuming, and soft-deleting agent sessions, plus Grok and Command Code sign-in from Docker.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.593 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.593
```

└──▷ TRY IT

Fetch the session list for a specific agent to audit, resume, or clean up old sessions programmatically.

```
$ curl 'http://localhost:<port>/api/sessions/manage?agent=claude-code'
```

Adds a Sessions page (`/api/sessions/manage`) that lists every agent's sessions by folder, lets you resume or delete them, and moves deleted sessions to `<magpie dir>/trash/sessions/` for later restore.

Supports Grok and Command Code sign-in from magpie running in Docker, installing Grok Build without requiring `bash` , `curl` , or `wget` in the container image.

v0.1.591 CODE ↗ STABLE

magpie v0.1.591 adds named gateway keys with per-client usage tracking, session-grouped request views, and `MAGPIE_PUBLIC_URL` support for reverse-proxy deployments.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.591 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.591
```

└──▷ TRY IT

Create a named key for a specific client so it can be revoked independently without affecting other clients.

```
$ magpie gateway-key add 'Workstation-Alice'
```

Rotate a compromised client key by ID while preserving its usage history and caller attribution.

```
$ magpie gateway-key rotate <id>
```

Adds `magpie gateway-key list` , `add <name>` , `rotate <id>` , and `remove <id>` subcommands for headless administration of named gateway keys; `list` masks credentials, while `add` / `rotate` print the fresh credential.

Tracks caller attribution via `callerKeyId` / `callerKeyName` (UI, filtering, CSV) and `caller_key_id` / `caller_key_name` (records) across chat, Responses, Messages, Gemini, images, video creation, streaming/failover, Codex own-model path, and `/v1/systemone` .

Introduces named gateway keys stored in `caller-keys.json` (prefixed `sk-magpie-key-`) with stable identity across rename/rotation, independent revocation or rotation per client, and enable/disable support; existing LAN credentials migrate to a key named `Magpie`.

Adds `MAGPIE_PUBLIC_URL` environment variable so the magpie web UI prints the correct shareable link when running behind a reverse proxy.

Adds a collapsible `By session` view in the Routing Requests list that groups calls by agent and explicit session ID, showing per-request and session cost estimates using the ledger's effective prices (including cache tiers and billable refusal retries); the grouping choice persists across reloads.

`v0.1.588` `CODE ↗` `STABLE`

magpie v0.1.588 persists conversation affinity across restarts and keeps Claude accounts alive after logout

`└──▷ GET THIS VERSION ▾`

```
$ git clone --branch v0.1.588 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.588
```

Persists conversation-to-account affinity in `affinity.json` (next to providers, up to 512 conversations) so routing survives a restart or in-place update without losing prompt-cache locality.

Keeps Claude accounts saved in magpie active and serving requests even while Claude Code is logged out, with no accounts dropped from the list.

Maintains per-account routing by each account's last known allowance until a fresh usage reading is available after startup.

Exposes a model's or group's `max` tier to oh-my-piper 16.4.0 and later, so routing groups with `max` enabled surface correctly instead of showing only `xhigh`.

`v0.1.587` `CODE ↗` `STABLE`

Magpie v0.1.587 adds bring-your-own search API fallback (Tavily, Brave, Exa, Firecrawl, SearXNG) when no provider can search.

`└──▷ GET THIS VERSION ▾`

```
$ git clone --branch v0.1.587 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.587
```

Adds `web_search` fallback to user-supplied search APIs — Tavily, Brave, Exa, Firecrawl, and SearXNG — via provider.SearchAPIs() when no configured provider can search natively.

Adds `magpie web` mode that skips requesting the Wails runtime in browser contexts, eliminating 404 console errors.

Improves the custom balance card to display the timestamp of the last read, each amount on its own line, and a percentage progress meter.

v0.1.585 CODE ↗ STABLE

magpie now self-manages Bun upgrades for its plugin host, with a 48-hour settle window and automatic fallback on failure.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.585 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.585
```

Automatically keeps the Bun runtime used by the plugin host up to date: every few hours magpie checks GitHub for the newest Bun release, waits 48 hours (`bunSettle`) before adopting it, verifies the checksum, and hot-swaps the host with no interruption to in-flight replies.

Tracks Bun versions in `bun/state.json` (fields: `current` , `previous` , `bad` , `checked`) so magpie can fall back to the previous working Bun if the new one fails to start the plugin host, and permanently set aside any version that crashes.

Respects the `MAGPIE_BUN` environment variable to pin a specific Bun binary and bypass automatic update logic entirely.

v0.1.584 CODE ↗ STABLE

Plugin providers now share full SOCKS5 proxy path with built-ins, and More subscriptions in Plugins moves to the add sheet.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.584 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.584
```

Plugin subscription providers now traverse the same proxy logic as built-in providers, including SOCKS5 proxy support via an internal HTTP-to-SOCKS5 bridge on loopback.

The More subscriptions in Plugins button is relocated to the add sheet (second page) rather than the providers list page.

v0.1.583 CODE ↗ STABLE

Plugins gain a folder picker via `POST /api/plugins/choose`, and plugin pages now show the folder's README instead of an npm error.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.583 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.583
```

└──▷ TRY IT

Open the system folder picker from a script or integration to let a user select a local plugin directory without typing a path.

```
$ curl -X POST http://localhost:<port>/api/plugins/choose
```

Adds `POST /api/plugins/choose` endpoint that triggers the system folder-picker panel (`Windows.ChooseFolder`) so plugins can be added by selecting a directory instead of typing a full path; disabled for `magpie web` (`webHost.ChooseFolder` refuses it), leaving the text field as the only input in browser-hosted mode.

Extends `plugin.Readme` to accept a filesystem path: `folderReadme` reads `README.md` (case-insensitive) or a neighbouring markdown/text file, so folder-installed plugins show their own README on their page instead of an npm error.

v0.1.581 CODE ↗ STABLE

magpie v0.1.581 adds wildcard model pricing via `*/<model>` and randomised Claude usage polling.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.581 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.581
```

└──▷ **TRY IT**

Set a provider-agnostic fallback price for a model that models.dev doesn't list, so usage from any gone or unknown provider is costed correctly.

```
$ magpie model price gemini-3-pro-preview 2,120,0,0
```

Adds `magpie model price <model> <in>,<out>,<cache read>,<cache write>` to set a fallback price stored as `*/<model>`, covering models whose provider is gone or that models.dev doesn't price (e.g. `gemini-3-pro-preview`).

v0.1.580 CODE ↗ STABLE

Magpie v0.1.580 unifies gateway and local session requests in a combined usage view with model charts, account labels, and CSV export.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.1.580 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.580
```

Adds a `rejected` column to the Requests detail table and CSV export, recording locally rejected requests without counting them in summaries, charts, or Sessions `via` field.

Combines gateway records with local session calls from Claude Code, Claude Desktop Code/Cowork, and Codex into a unified Requests view with token, request count, cost, and cache statistics.

Adds per-request time-series charts defaulting to a per-model breakdown in the Requests view.

Adds account labels to the Requests view with distinct badges distinguishing official login, API, subscription, and local-session sources.

Changes the default usage window from `30d` to `today` across the usage overview and quota panel.

+3 more

v0.1.578 CODE ↗ STABLE

magpie v0.1.578 adds GET `/v1`/magpie/route to expose live session routing before the first token arrives.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.578 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.578
```

Automatic sign-back-in to the first-created account once its usage share drops below `backShare` (90%), preventing Codex and Claude Code from staying on a secondary account after the primary recovers.

v0.1.577 CODE ↗ STABLE

Magpie v0.1.577 adds S3 request archiving with redaction, group fast-mode routing, and unified usage analytics across gateway and local sessions.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.577 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.577
```

└──▷ USE IT

Enable request archiving so every gateway call is preserved in your S3 bucket for audit or replay — secrets are automatically redacted before upload.

yaml

```
RequestArchive: true
```

Adds `RequestArchive` setting that stores each gateway call's headers and bodies (first 256 KB, secrets redacted) in the user's S3 bucket at `<prefix>/magpie/archive/<date>/<id>.json`, with the archive ID returned on responses via the `X-Magpie-Archive-Id` header.

Adds support for sending a routing group member using its vendor's fast mode alongside its own reasoning, enabling mixed effort-level dispatch within a single group.

Marks Command Code free models as free and folds disabled providers below enabled ones in the provider list.

v0.1.574 CODE ↗ STABLE

Magpie gains Claude account sign-in via `claude auth login`, with magpie acting as the browser to capture and complete the OAuth flow.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.574 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.574
```

v0.1.573 CODE ↗ STABLE

Magpie v0.1.573 adds automatic background updates for community plugins with a dot-indicator for third-party ones.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.573 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.573
```

Community plugins (`@magpie-community/*`) now update automatically in the background shortly after magpie starts and every few hours, without reader intervention.

Third-party plugins and version-pinned plugins surface a dot on the Plugins screen when a newer version is available, letting the reader update with a single click.

Plugin updates are non-disruptive: an in-progress streaming reply or pending sign-in finishes on the old plugin host before the new version takes over.

v0.1.572 CODE ↗ STABLE

magpie v0.1.572 links usage rows to routing history, adds Grok namespace tool support, and improves GLM thinking streams.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.572 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.572
```

Adds `route_id` field to `usage.Record`, included in CSV exports, linking each usage row (including refusals and Codex pass-through requests) to its gateway routing story.

Adds exact route filter support in the ledger API, enabling lookup of all usage rows for a specific route.

Enables Codex sub-agent tools (`spawn_agent` , `wait_agent` , `send_message` , `followup_task` in the `collaboration` namespace, and `set_thread_title` in `codex_app`) to pass through to a Grok subscription by translating namespaced tools to flat names (e.g. `collaboration__spawn_agent`).

Adds Claude Desktop to the ui-preview sandbox alongside Claude Code, Codex, Gemini, and OpenCode.

Shows a UI indicator when a model is reserved for routing groups only, rather than silently removing it from agent model lists.

v0.1.570 CODE ↗ STABLE

Adds time-of-day routing windows (`time: {from, to, days}`) to group rules, plus ZCode token-count estimation and blocked-request messaging.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.570 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.570
```

└──▷ USE IT

Route DeepSeek or GLM requests to a cheaper provider during peak-price hours on weekday evenings, falling back after midnight.

yaml

```yaml
time:
  from: "22:00"
  to: "08:00"
  days:
    - "mon"
    - "tue"
    - "wed"
    - "thu"
    - "fri"
```

Adds `time` field (a `TimeWindow` with `from` , `to` , and `days` keys, each in `HH:MM` format) to routing group rules, so requests arriving during a vendor's peak-price hours are redirected to another provider on the local machine's clock.

Supports overnight windows in `TimeWindow` when `to` comes before `from` (e.g. `22:00` – `08:00`), spanning past midnight on the day the window opens.

ZCode `count_tokens` calls are now estimated locally rather than forwarded to `zcode.z.ai`, which always rejects them with an error.

Surfaces ZCode Start Plan firewall blocks (HTTP 405) as a plain-language message instead of a raw error.

`v0.1.569` `CODE ↗` `STABLE`

Magpie v0.1.569 shows Copilot allowance cards for magpie-signed-in accounts and preserves plugin model lists during vendor outages.

`└──▷ GET THIS VERSION ▾`

```
$ git clone --branch v0.1.569 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.569
```

Adds `FellBack` field (`fellBack`) to `Provider` and `Account` JSON objects so callers can detect when a plugin fell back to its cached model list instead of a freshly fetched one.

Copilot allowance cards now appear for accounts signed in via magpie itself, even when no editor or Copilot CLI sign-in exists on the machine (`~/.config/github-copilot/` or `~/.copilot/config.json` no longer required).

Plugin model lists now survive vendor fetch failures by retaining the last successfully fetched list rather than falling back to the plugin's short built-in defaults.

`v0.1.568` `CODE ↗` `STABLE`

magpie v0.1.568 tracks which upstream provider key served each call, with new `providerKeyId` / `providerKeyName` fields across JSON, CSV, and Go.

`└──▷ GET THIS VERSION ▾`

```
$ git clone --branch v0.1.568 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.568
```

Adds `providerKeyId` and `providerKeyName` fields to JSON usage records, `provider_key_id` and `provider_key_name` columns to CSV usage records, and a `providerKeys` summary field to track which upstream credential actually served each request.

Adds an upstream provider keys table to `magpie usage` output, aggregated by provider plus key fingerprint, covering key failover, billed refusals, images, `/v1/systemone` calls, and router-initiated System One calls.

Preserves historical key identity (name and fingerprint) after key removal, rotation, or rename — old records remain labeled `key not recorded` rather than being reassigned to the current key.

`v0.1.567` `CODE ↗` `STABLE`

Gateway stream decoder now correctly continues an in-progress tool call when a relay repeats the call's `id` on every fragment, rather than opening a new call each time.

`└──▷ GET THIS VERSION ▾`

```
$ git clone --branch v0.1.567 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.567
```

Streamed tool-call handling in the gateway correctly continues an existing call when a relay repeats the tool call `id` on every fragment, delivering a single complete call with valid JSON arguments.

`v0.1.561` `CODE ↗` `STABLE`

Magpie v0.1.561 adds a structured SSE viewer that parses streamed reply bodies into named events instead of showing raw event: `/data` : lines.

`└──▷ GET THIS VERSION ▾`

```
$ git clone --branch v0.1.561 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.561
```

Adds a server-sent events (SSE) body viewer in the gateway recent-calls panel, rendering streamed replies as parsed events (name + data) rather than raw `event:` / `data:` lines, with a `magpie.sseView` localStorage preference defaulting to `'events'` .

Introduces three view modes for streamed response bodies — `Reply` , `Events` , and `Raw` — selectable per call and persisted across sessions via `magpie.sseView` .

`v0.1.560` `CODE ↗` `STABLE`

magpie v0.1.560 adds `magpie model wire` to send provider-specific upstream model names on the wire without renaming the canonical ID.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.560 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.560
```

└──▷ TRY IT

When a relay serves a model under a dated or prefixed name the provider doesn't recognise by its canonical ID, pin the wire name so requests succeed without renaming the model everywhere.

```
$ magpie model wire relay-b/model-2 vendor-c/model-2-preview
```

When every model on a provider needs to be requested under a vendor-specific name, use the wildcard form to set a provider-wide override in one step.

```
$ magpie model wire relay-b/* vendor-c/model-2
```

When a provider-specific wire name is no longer needed, remove the exact override so requests revert to the canonical model ID.

```
$ magpie model wire relay-b/model-2 --reset
```

Adds `magpie model wire <provider/model> <upstream-id>` subcommand to override the name sent to a provider on the wire, with a `<provider>/*` wildcard form for provider-wide overrides and `--reset` to remove an exact override.

v0.1.553 CODE ↗ STABLE

magpie v0.1.553 adds `MAGPIE_PUBLIC_URL` for Docker LAN sharing, per-plugin icons, agent attribution on Remote magpie, and skippable update prompts.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.553 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.553
```

└──▷ **TRY IT**

Run magpie in Docker so other machines on your LAN can reach it by your host's real IP instead of the container-internal address.

```
$ docker run -e MAGPIE_PUBLIC_URL=http://192.168.1.42:9090 -p
9090:9090 yetone/magpie
```

Supports `MAGPIE_PUBLIC_URL` environment variable so a magpie instance running in Docker advertises a reachable LAN address to other computers instead of its internal container address.

Plugins can now declare their own icon and customize the API key field title and hint, overriding the marketplace defaults.

Remote magpie now records the originating agent on the receiving instance, so per-agent usage tracking is accurate across multi-machine setups.

Codex compact mode now reaches xAI, Grok, and Copilot providers, and web-search turns preserve `client_metadata`.

Sign-ins and plugins now persist across Docker container restarts, and the Docker image includes a built-in health check.

+1 more

v0.1.552 CODE ↗ STABLE

magpie v0.1.552 surfaces plugin move/migrate commands in `--help`, improves migration UX with failure reasons, and fixes provider display after built-in moves.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.1.552 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.552
```

Adds `plugin move|migrate` and `move-back|unmigrate` subcommands to `magpie --help` so the full migration surface is discoverable from the top-level help.

Migration failures now return a structured `MoveWhy` object (with `code` and `args`

fields) so the GUI can explain why a move did not go through in the user's language.

A failed move leaves nothing behind and only proceeds when the plugin has verified a working account, preventing partial migrations.

`magpie plugin list` now shows that a built-in is running via its plugin's provider instead of displaying `not signed in` after a move.

`magpie accounts` and the Providers view preserve a moved built-in's position and display the vendor's host rather than the plugin's id.

+1 more

v0.1.551 CODE ↗ STABLE

magpie v0.1.551 adds per-provider `searches` flag for relay passthrough and expands omp model picker to read `~/.omp/agent/models.yml`

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.551 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.551
```

└──▷ TRY IT

Tell magpie that a custom Anthropic-compatible relay performs its own web search, so Claude Code's `web_search` requests are passed through unchanged rather than rebuilt.

```
$ magpie provider set my-relay search=yes
```

Verify the relay's search flag is recognised — output should show `search by itself` in the provider detail.

```
$ magpie provider my-relay
```

Adds `searches` field (bool) to `provider.Provider` and `search=yes|no` argument to `magpie provider add/set <id>` so a relay that performs its own web search receives Claude Code's `web_search` request as-is instead of being rebuilt.

Adds ompOwnOptions(modelsFile, cur) in `internal/agent/omp.go` so omp's model picker reads `~/.omp/agent/models.yml` (falling back to `models.yaml`) and lists every `provider/id` from every provider defined there, including gateways and relays unknown to models.dev.

Removes `claude_cloak.go` (346 lines deleted): magpie no longer impersonates Claude Code's wire identity when sending requests to Anthropic, and a Claude account's model test now runs via the CLI instead.

Adds a `Web search — searches by itself` toggle in the custom provider editor UI (shown only when an Anthropic or Responses endpoint address is set), with a note that magpie will not use this provider to search for other models.

└──▷ **BREAKING ON UPGRADE**

! magpie no longer adds Claude Code headers (User-Agent, anthropic-beta, X-Claude-Code-Session-Id, etc.) when signing direct requests to Anthropic on behalf of a Claude account; p.Sign() now returns `errClaudeViaCLI` and sets no Authorization header for such requests. Any workflow that relied on magpie forwarding signed Claude Code identity headers to Anthropic directly will stop working.

v0.1.544 CODE ↗ STABLE

Adds image generation and video creation through Grok subscriptions (SuperGrok/X Premium+) via the Grok Imagine API

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.1.544 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.544
```

└──▷ **TRY IT**

Start a video generation job through the gateway using an OpenAI-compatible SDK or curl — the async design means you get an ID back immediately and poll for completion.

```
$ curl -X POST https://<magpie-gateway>/v1/videos \
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "grok-imagine-video", "prompt": "a fox running
through a snowy forest at dusk"}'
```

Poll for the finished video once you have the job ID returned by the creation request.

```
$ curl https://<magpie-gateway>/v1/videos/<id> \
  -H 'Authorization: Bearer <token>'
```

Adds `POST /v1/videos`, `GET /v1/videos/{id}`, and `GET /v1/videos/{id}/content` endpoints to the gateway, implementing OpenAI's Videos API shape so existing OpenAI SDKs work unchanged against a Grok subscription.

Adds `grok-imagine-video` and `grok-imagine-video-1.5` as video models routed through the Grok Build backend at `cli-chat-proxy.grok.com/v1/videos/generations`.

Adds a `generate_video` MCP tool to `magpie mcp image`, enabling agents to generate short videos from a text prompt or a starting image using a Grok subscription; videos are saved to the `generated-videos` folder.

Adds image generation and editing for Grok subscriptions (SuperGrok, X Premium+) via `cli-chat-proxy.grok.com/v1/images/generations` and `/images/edits`, exposing `grok-imagine-image` and `grok-imagine-image-quality` models.

Adds `generate_image` MCP tool support for Grok-backed image generation, with generated images saved to the `generated-images` folder.

WAS THIS USEFUL? 👍 👎

OpenAI Codex CLI

Feedback ⓘ rust-v0.161.0-alpha.8 NOTES ↗

CODE ↗ RANK

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

> Codex CLI added a persistent Daybreak mode spanning the TUI and `codex exec`, per-channel microphone selection for voice conversations, default-on API-key model discovery, and AWS GovCloud support for Amazon Bedrock, alongside reliability and UI refinements to the exec server, launch permissions, and diagnostics.

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

THINNER COVERAGE BELOW

10 **Account security setup reminders** NEW 25

Eligible local sessions signed in with ChatGPT can now display optional reminders to complete account security setup.

— Thin description with no mechanism or configuration named rust-v0.159.3

rust-v0.159.3 NOTES ↗ CODE ↗ LATEST STABLE

Codex CLI v0.159.3 adds optional account security setup reminders for local ChatGPT sessions.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch rust-v0.159.3
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.159.3
```

WAS THIS USEFUL? 👍 👎

Cline

Feedback ⓘ desktop-v0.0.40 NOTES ↗ CODE ↗ RANK

Autonomous coding agent as an SDK, IDE extension, or CLI assistant.

> Cline's biggest change this window is a new way to handle oversized MCP tool output — paged previews backed by an in-memory cache and `cline://cache/` URIs — alongside fixes to custom provider execution, a refreshed model catalog, and breaking changes to an SDK function signature and Vultr model IDs.

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

01 **Paging through oversized MCP tool output** NEW 95

When an MCP tool result exceeds context, the agent now returns a bounded preview plus a navigable link instead of truncating. Custom tools can opt into this via `resultPolicy: "cache-oversized"` in `createTool`, which stores the oversized result in a per-session in-memory cache and returns a `cline://cache/...` URI that `read_files` can page through by line range.

Register a custom tool that caches large outputs so the agent can page through them without hitting context-window limits.

typescript

```
import { Agent, createTool } from "@cline/sdk"

const bigResultTool = createTool({
  name: "fetch_logs",
  description: "Fetch all application logs for a time range.",
  inputSchema: { type: "object", properties: { since: { type:
"string" } }, required: ["since"] },
  resultPolicy: "cache-oversized",
  execute: async (input) => {
    // returns potentially large log output
    return fetchLogsFrom(input.since)
  },
})

const agent = new Agent({ tools: [bigResultTool] })
```

— Full mechanism plus a runnable SDK example sdk/sdk/v0.0.89 desktop-v0.0.40

02 **Unified MCP settings path via env var** NEW 80

The `CLINE_MCP_SETTINGS_PATH` environment variable points all MCP settings operations — server list, OAuth, and the settings UI — at a shared path across machines or containers.

Point all MCP settings operations — server list, OAuth, and the settings UI — at a shared path across machines or containers.

```env
CLINE_MCP_SETTINGS_PATH=/shared/config/mcp-settings.json
```

— Exact env var with a runnable example `desktop-v0.0.40`

03 **SDK exports for custom provider registration** NEW 75

`@cline/llms` now exports `resolveGatewayProviderRegistration` and `resolveGatewayProviderRegistrationSync` so custom providers defined in `providers.json` / `models.json` can run on the agent path.

— Exact export names and config files, no usage example `sdk/sdk/v0.0.89`

04 **saveLocalProviderSettings now async** BREAKING 70

`saveLocalProviderSettings` is now async — callers that previously invoked it synchronously must be updated to `await` the result.

— Exact function name with clear migration instruction `sdk/sdk/v0.0.89`

05 **Vultr model ID renames** BREAKING 70

Vultr model IDs were renamed upstream (e.g. `zai-org/GLM-5.2-FP8` is now `glm-5.2`); any pinned Vultr model ID in provider configuration may need to be re-selected after upgrading.

— Gives old/new IDs and a clear remediation `sdk/sdk/v0.0.89` `cli-v3.0.67`

THINNER COVERAGE BELOW

06 **Custom provider execution fixed** IMPROVED 50

Custom providers added via Add Provider now execute correctly in tasks instead of failing with Unknown or disabled provider.

— Names the fixed error but no reproduction steps `desktop-v0.0.40`

07 **Chat remembers Cloud/Local model selection** IMPROVED 40

New chats remember the last-used Cloud or Local selection and the specific Cloud model chosen, rather than always reverting to Local.

— Describes behavior change with no config surface `desktop-v0.0.40`

└──▷ BREAKING ON UPGRADE

! Vultr model IDs were renamed upstream; a pinned Vultr model may need to be re-selected after upgrading.

! `saveLocalProviderSettings` is now async — callers that previously called it synchronously must be updated to `await` the result.

! Vultr model IDs were renamed upstream (e.g. `zai-org/GLM-5.2-FP8` is now `glm-5.2`); any pinned Vultr model ID in provider configuration may need to be re-selected.

`desktop-v0.0.40` NOTES ↗ CODE ↗ LATEST STABLE

Cline desktop v0.0.40 fixes custom providers, unifies MCP settings paths via env vars, and enables paging through oversized tool output.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch desktop-v0.0.40
https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout desktop-v0.0.40
```

└──▷ USE IT

Point all MCP settings operations — server list, OAuth, and the settings UI — at a shared path across machines or containers.

.env

```
CLINE_MCP_SETTINGS_PATH=/shared/config/mcp-settings.json
```

Enables the agent to page through oversized MCP tool output that exceeds context — returns a preview plus a navigable link instead of truncating.

Custom providers added via `Add Provider` now execute correctly in tasks instead of failing with `Unknown or disabled provider`.

New chats remember the last-used `Cloud` or `Local` selection and the specific Cloud model chosen, rather than always reverting to Local.

`sdk/sdk/v0.0.89` NOTES ↗ CODE ↗ STABLE

Cline SDK v0.0.89 adds oversized MCP tool result caching with `cline://cache/` URIs and `resultPolicy: "cache-oversized"` for custom tools.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch sdk/sdk/v0.0.89
https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout sdk/sdk/v0.0.89
```

└──▷ USE IT

Register a custom tool that caches large outputs so the agent can page through them without hitting context-window limits.

```typescript
import { Agent, createTool } from "@cline/sdk"

const bigResultTool = createTool({
  name: "fetch_logs",
  description: "Fetch all application logs for a time range.",
  inputSchema: { type: "object", properties: { since: { type:
"string" } }, required: ["since"] },
  resultPolicy: "cache-oversized",
  execute: async (input) => {
    // returns potentially large log output
    return fetchLogsFrom(input.since)
  },
})

const agent = new Agent({ tools: [bigResultTool] })
```

Adds `resultPolicy: "cache-oversized"` option to `createTool` so custom tools can opt oversized results into a per-session in-memory cache, with the model receiving a bounded preview plus a `cline://cache/...` URI that `read_files` can page through by line range.

Exports `resolveGatewayProviderRegistration` and `resolveGatewayProviderRegistrationSync` from `@cline/llms` so custom providers defined in `providers.json` / `models.json` can run on the agent path.

Refreshes the model catalog: adds GPT-6.1 Sol to the Cline recommended list, drops Pixel Canary, and updates defaults for 302.AI (Claude Sonnet 5.5), NanoGPT (Ling 3.1 Flash), Vivgrid (GPT-6.1 Sol), and Vultr (MiMo V2.6 Flash RL).

└──▷ BREAKING ON UPGRADE

- **!** `saveLocalProviderSettings` is now async — callers that previously called it synchronously must be updated to `await` the result.
- **!** Vultr model IDs were renamed upstream (e.g. `zai-org/GLM-5.2-FP8` is now `glm-5.2`); any pinned Vultr model ID in provider configuration may need to be re-

selected.

WAS THIS USEFUL? 👍 👎

GitHub Copilot CLI

Feedback ⓘ v1.0.90 NOTES ↗ RANK

GitHub Copilot CLI is a terminal-based coding agent that can plan, write, and test code in local repositories.

> GitHub Copilot CLI v1.0.90 adds scoped MCP authentication and session-level directory read approvals to tighten control over credential sharing and path access prompts.

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

01 **Scoped GitHub auth for MCP servers** NEW 80

The new `--mcp-github-auth` flag scopes GitHub account authentication to approved MCP server origins, preventing unintended credential sharing with arbitrary MCP servers. Example usage: `copilot --mcp-github-auth`.

Launch Copilot CLI with GitHub auth scoped only to approved MCP server origins, preventing unintended credential sharing with arbitrary MCP servers

```
$ copilot --mcp-github-auth
```

— Names exact flag and purpose, with runnable example v1.0.90

THINNER COVERAGE BELOW

02 **Session-scoped read-only directory approvals** NEW 45

Path access prompts now support session-scoped read-only directory approvals, letting users approve directory reads for the duration of a session rather than per-request.

— Describes behavior but no flag or command given v1.0.90

v1.0.90 NOTES ↗ LATEST STABLE

Adds GPT-6.1 Sol model, `--mcp-github-auth` flag for scoped MCP auth, and session-scoped read-only directory approvals

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.0.90 https://github.com/github/copilot-cli.git
# already have the repo? check out this version:
$ git checkout v1.0.90
```

└──▷ TRY IT

Launch Copilot CLI with GitHub auth scoped only to approved MCP server origins, preventing unintended credential sharing with arbitrary MCP servers

```
$ copilot --mcp-github-auth
```

Adds `--mcp-github-auth` flag to scope GitHub account authentication to approved MCP server origins

Adds session-scoped read-only directory approvals to path access prompts, letting users approve directory reads for the duration of a session

---

WAS THIS USEFUL? 👍 👎

Graphify Labs Graphify

Feedback ⓘ v0.9.73 NOTES ↗

RANK

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

> Graphify v0.9.73 expands its code-parsing coverage with enum extraction across four more languages and adds support for Solidity's file-level free functions.

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

01 **Enum extraction across Rust, Zig, C++, Scala 3** NEW

68

Enum members are now extracted as nodes with `case_of` edges in Rust (enum variants), Zig (enum members), C++ (`enum` / `enum class` enumerators including nested enums), and Scala 3 (enum cases plus their methods).

— Names exact languages and edge type, no usage example v0.9.73

THINNER COVERAGE BELOW

02 **Solidity free-function graph support** NEW

59

Extracts Solidity file-level free functions (Solidity 0.7+, declared outside any contract) and their calls as graph nodes and edges.

— Names version and mechanism but no example or command v0.9.73

v0.9.73 NOTES ↗ LATEST STABLE

Graphify v0.9.73 adds enum extraction for Rust, Zig, C++, and Scala 3, plus Solidity free-function support.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.9.73 https://github.com/Graphify-Labs/graphify.git
# already have the repo? check out this version:
$ git checkout v0.9.73
```

Extracts enum members as nodes with `case_of` edges in Rust (enum variants), Zig (enum members), C++ (`enum` / `enum class` enumerators including nested enums), and Scala 3 (enum cases plus their methods).

Extracts Solidity file-level free functions (Solidity 0.7+, declared outside any contract)

> and their calls as graph nodes and edges.

WAS THIS USEFUL? 👍 👎

Earendil Works Pi

Feedback ⓘ v0.99.2 NOTES ↗ RANK

AI agent toolkit: unified LLM API, agent loop, TUI, coding agent CLI

> Pi's latest releases focus on MCP server authentication and tool discovery: new OAuth client-name overrides, provider-based bearer-token auth, and Anthropic workload identity federation sit alongside a new MCP server description field, more flexible namespace lookups, and changes to codemode tool exposure — plus breaking changes to system-prompt formatting and MCP tool naming.

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

01 Provider bearer-token auth for HTTP MCP servers NEW

83

The `"auth": { "provider": "<provider>" }` config for HTTP MCP servers sends the provider's current `/login` token as the bearer token on every request in place of MCP OAuth; it requires HTTPS except on loopback hosts, and is only allowed in the global `mcp.json` and from extensions.

Authenticate an HTTP MCP server using a provider's login token instead of MCP OAuth — useful when the server expects a pre-existing bearer token from your SSO provider.

```json
{"mcpServers": {"my-server": {"url": "https://internal.example.com/mcp", "auth": {"provider": "my-provider"}}}}
```

Configure an HTTP MCP server to authenticate using a provider's bearer token instead of MCP OAuth — useful when the server expects a short-lived token from an existing provider.

```json
{
  "auth": { "provider": "anthropic" }
}
```

— Names config key, scope restriction, HTTPS requirement, with example config v0.99.2

02 Anthropic workload identity federation NEW

78

Pi adds Anthropic workload identity federation via the `ANTHROPIC_FEDERATION_RULE_ID`, `ANTHROPIC_ORGANIZATION_ID`, and `ANTHROPIC_IDENTITY_TOKEN_FILE` environment variables, letting Pi pick up credentials from the environment without a static API key.

Enable Anthropic workload identity federation so Pi authenticates without a static API key, e.g. in a CI environment with federated credentials.

```
$ export ANTHROPIC_FEDERATION_RULE_ID=rule-abc123
export ANTHROPIC_ORGANIZATION_ID=org-xyz
export
ANTHROPIC_IDENTITY_TOKEN_FILE=/run/secrets/anthropic_token
pi
```

Enable Anthropic workload identity federation so Pi picks up credentials from the environment without a static API key.

```
$ export ANTHROPIC_FEDERATION_RULE_ID=my-rule-id
export ANTHROPIC_ORGANIZATION_ID=my-org-id
export ANTHROPIC_IDENTITY_TOKEN_FILE=/run/secrets/identity_token
pi <command>
```

— Three named env vars with runnable export commands v0.99.2

03 **Description field for MCP servers** NEW 75

The `--description <text>` flag (used via `pi mcp add --description`) sets a description for an MCP server, shown in the system prompt and used to rank tools in tool search; the flag can also be used standalone to print a description without establishing an MCP connection.

Print the CLI's description without establishing an MCP server connection — useful for quickly auditing what tools will be advertised.

```
$ pi --description 'Security scanning assistant' --oauth-client-
name my-client
```

— Names exact flag and command but mechanism of ranking not detailed v0.99.2

04 **Flexible namespace lookup in describeNamespace and searchTools** NEW 70

New describeNamespace(name) codemode helper returns a namespace's instructions and tool names. Both describeNamespace() and searchTools() now accept a namespace in hyphenated or underscored form (`mcp__dev-radius` , `mcp__dev_radius` , `dev-radius` , `dev_radius`).

— Named API functions and accepted formats, but no runnable command shown v0.99.2

05 **OAuth client name override for MCP servers** NEW 70

The `oauth.clientName` setting, set via `pi mcp add --oauth-client-name`, lets an MCP server override the client name sent during OAuth registration, for servers that only accept known clients.

Register a private MCP server that only accepts a known OAuth client name, so Pi presents the right identity during OAuth registration.

```
$ pi mcp add https://internal.example.com/mcp --oauth-client-name my-approved-client
```

Register an MCP server that requires a known OAuth client name during client registration.

```
$ pi mcp add --oauth-client-name my-registered-client https://my-mcp-server.example.com
```

— Exact flag and command given with working example `v0.99.2`

---

06 **MCP tool and namespace name hyphen/underscore normalization** 62

BREAKING

MCP tool and namespace names now replace `-` with `_` (e.g. `mcp__my-server__x` becomes `mcp__my_server__x`); codemode scripts referencing the old hyphenated names will call the wrong tool or fail. Server names that differ only in `-` vs `_` are now rejected, and colliding MCP tool names (tools of the same server differing only in `-` and `_`) get a hash suffix appended.

— Clear before/after example but no migration command given `v0.99.2`

---

07 **Codemode tool exposure behavior changes** IMPROVED 60

MCP servers with default `codemode` exposure are no longer listed in the `codemode` description and no longer block the first prompt; they connect in the background and are waited for only when needed. `codemode-deferred` is now an alias for `codemode`; use `direct` exposure for tools the model should see without searching.

— Explains before/after behavior but gives no command to try it `v0.99.2`

THINNER COVERAGE BELOW

---

08 **Mid-session tool reload via /reload** IMPROVED 55

`/reload` now enables tools newly added to the `tools` setting mid-session, while respecting `--no-tools` and `--no-builtin-tools` overrides.

— Names command and overrides but no worked example `product docs`

09 **System prompt description no longer includes deferred tool details**

55

BREAKING

The system prompt `description` no longer includes deferred tools, tool counts, or MCP server instructions — scripts that parsed those fields will need updating. Servers whose tools it can load now appear in a separate `mcp_servers` system prompt section updated at the start of each prompt.

— States exact before/after field changes but no migration command product docs

└──▷ BREAKING ON UPGRADE

! The system prompt `description` no longer includes deferred tools, tool counts, or MCP server instructions — scripts that parsed those fields will need updating.

! The system prompt `description` no longer lists servers whose tools it can load; servers now appear in a separate `mcp_servers` system prompt section updated at the start of each prompt.

! MCP tool and namespace names now replace `-` with `_` (e.g. `mcp__my-server__x` becomes `mcp__my_server__x`); codemode scripts referencing the old hyphenated names will call the wrong tool or fail.

! Server names that differ only in `-` vs `_` are now rejected.

! Colliding MCP tool names (tools of the same server that differ only in `-` and `_`) now get a hash suffix appended.

v0.99.2 NOTES ↗ LATEST STABLE

Pi v0.99.2 adds MCP auth options, Anthropic workload identity federation, and smarter tool exposure controls.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.99.2 https://github.com/earendil-works/pi.git
# already have the repo? check out this version:
$ git checkout v0.99.2
```

└──▷ TRY IT

Register a private MCP server that only accepts a known OAuth client name, so Pi presents the right identity during OAuth registration.

```
$ pi mcp add https://internal.example.com/mcp --oauth-client-name my-approved-client
```

Authenticate an HTTP MCP server using a provider's login token instead of MCP OAuth — useful when the server expects a pre-existing bearer token from your SSO provider.

mcp.json

```
{"mcpServers": {"my-server": {"url":
"https://internal.example.com/mcp", "auth": {"provider": "my-
provider"}}}}
```

Enable Anthropic workload identity federation so Pi authenticates without a static API key, e.g. in a CI environment with federated credentials.

```
$ export ANTHROPIC_FEDERATION_RULE_ID=rule-abc123
export ANTHROPIC_ORGANIZATION_ID=org-xyz
export ANTHROPIC_IDENTITY_TOKEN_FILE=/run/secrets/anthropic_token
pi
```

Adds `oauth.clientName` setting (`pi mcp add --oauth-client-name`) for MCP servers to override the client name sent during OAuth registration, for servers that only accept known clients.

Adds `"auth": { "provider": "<provider>" }` config for HTTP MCP servers to send a provider's current `/login` token as the bearer token on every request, in place of MCP OAuth; requires HTTPS except on loopback hosts, and is only allowed in the global `mcp.json` and from extensions.

Adds Anthropic workload identity federation via `ANTHROPIC_FEDERATION_RULE_ID` , `ANTHROPIC_ORGANIZATION_ID` , and `ANTHROPIC_IDENTITY_TOKEN_FILE` environment variables.

Adds `description` field for MCP servers (`pi mcp add --description`) shown in the system prompt and used to rank tools in tool search.

Adds describeNamespace(name) codemode helper that returns a namespace's instructions and tool names; accepts namespace as `mcp__dev-radius` , `mcp__dev_radius` , `dev-radius` , or `dev_radius` .

+2 more

└─▷ **BREAKING ON UPGRADE**

! MCP tool and namespace names now replace `-` with `_` (e.g. `mcp__my-server__x` becomes `mcp__my_server__x`); codemode scripts referencing the old hyphenated names will call the wrong tool or fail.

! Server names that differ only in `-` vs `_` are now rejected.

! Colliding MCP tool names (tools of the same server that differ only in `-` and `_`) now get a hash suffix appended.

WAS THIS USEFUL? 👍 👎

Anthropic Claude Code

Feedback ⓘ v2.1.286 NOTES ↗

CODE ↗ RANK

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

> Claude Code v2.1.286 reworks permission-prompt and fullscreen list-screen UI, expands the VSCode extension with bookmarks, question history and richer message context, tightens plugin install sources and retry behavior, and adds commit-time skill guidance.

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

01 **Fullscreen list navigation and layout overhaul** IMPROVED 75

List screens such as `/artifacts`, `/mcp`, `/skills` and `/hooks` now always line up each row's details in one column after the names, and `/hooks` opens on a single list of configured hooks grouped by event so viewing one takes one Enter instead of three. The N more overflow rows in fullscreen lists now read `↑ N more` / `↓ N more`, support mouse clicks with hover and pressed states to jump to that end of the list, and have scrollbars that no longer shift as rows appear plus clickable/holdable ↑/↓ arrows. The theme picker is now a scrolling list sized to the terminal instead of pushing the preview off screen, and number keys no longer select a theme.

— Names exact commands and overflow text strings v2.1.286

02 **Unified retry budget per model call** IMPROVED 70

Failed API request retries are now capped by a single limit covering a whole model call rather than per-request retries; with the default retry settings a failing call now sends at most 14 requests.

— Gives exact retry cap number and mechanism v2.1.286

03 **Automatic verify-skill guidance before commits** IMPROVED 67

When a project or user skill named `verify` is installed, Claude Code now tells Claude to run it right before committing, except for docs-only and tests-only commits.

— Names exact skill name and commit exceptions v2.1.286

04 **Permission prompt UI improvements** IMPROVED 62

Permission prompts now show a count like `2 of 5` when several requests stack up. The fetch, skill, file read, sandbox network, Claude in Chrome, workflow script and

notebook edit prompts have been restyled to match file edit prompts, and Bash, PowerShell and Monitor prompts now show the command between dashed lines.

— Names every affected prompt type but UI-only change v2.1.286

05 **Plugin install source restrictions tightened** BREAKING 60

Plugin installs now refuse npm sources that are git repositories or folders, and install plugin dependencies only from registry packages.

— Names exact restricted source types v2.1.286

THINNER COVERAGE BELOW

06 **Send-now key backgrounds skill shell commands** IMPROVED 55

The send-now key (`ctrl+enter`) now moves a skill's own shell command to the background instead of ending it.

— Names exact keybinding and new behavior v2.1.286

07 **Bookmarks panel in VSCode extension** NEW 50

Adds VSCode bookmarks: you can save Claude's responses and keep them visible in a dedicated Bookmarks side panel.

— Describes UI location but no further mechanism v2.1.286

08 **External editor opens at cursor line** IMPROVED 50

External editors invoked with Ctrl+G that support line numbers now open directly on the line your cursor was on in the prompt.

— Names exact shortcut and new behavior v2.1.286

09 **Expandable context rows under VSCode messages** NEW 48

Adds rows under a VSCode message that open to the terminal output, browser tab, browser instructions, and selected code that were sent with that message.

— Names four context types surfaced per message v2.1.286

10 **Question history shown in VSCode chat** NEW 45

After you answer a question card in the VSCode extension, a Questions row is added to the conversation showing each question that was asked and the answer you picked.

— Clear behavior described, no config surface v2.1.286

11 **Live previews for question card options** NEW 45

Question cards in the VSCode chat panel now show the highlighted choice's mockup or code snippet beside or under the answer options.

— Behavior described but no further detail v2.1.286

12 **Faster, prefix-matched slash command suggestions** IMPROVED 45

Slash command suggestions now stay responsive while typing even with many installed skills or plugin commands, and command descriptions now match by word prefix instead of requiring an exact start.

— Describes matching change without exact examples v2.1.286

13 **Browser link removed from ultrareview output** DEPRECATED 40

Removes the browser link from `/ultrareview` and `claude ultrareview` output.

— Names exact commands but minimal explanation v2.1.286

14 **Add-channel button for Claude Tag spend limits** NEW 37

Adds an Add channel button to Claude Tag's spend limits page in admin settings.

— Names exact admin page, minimal further detail v2.1.286

15 **VSCode status bar item always visible** IMPROVED 35

The ✻ Claude Code status bar item now shows in every VSCode window, so you can open Claude when no file is open.

— Simple UI tweak with no further mechanism v2.1.286

v2.1.286 NOTES ↗ CODE ↗ LATEST STABLE

Claude Code v2.1.286 adds VSCode bookmarks, permission-prompt stacking counts, mouse nav in fullscreen lists, and a hardened `--bare` mode.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.1.286 https://github.com/anthropics/claude-code.git
# already have the repo? check out this version:
$ git checkout v2.1.286
```

Adds a count such as 2 of 5 to the permission prompt when several permission requests stack up.

> Adds mouse support for the `N more` rows of lists in fullscreen mode: click one to jump to that end of the list, with hover and pressed states.
>
> Adds VSCode bookmarks: save Claude's responses and keep them in view in a Bookmarks side panel.
>
> Adds the questions Claude asks and your answers to the VSCode conversation: after you answer a question card, a Questions row shows each question with your picks.
>
> Adds option previews to question cards in the VSCode chat panel: the highlighted choice's mockup or snippet shows beside or under the options.
>
> `+18 more`

WAS THIS USEFUL? 👍 👎

Zed

Feedback ⓘ v1.22.0 NOTES ↗ RANK

Zed is an open-source code editor that runs collaborative development workflows.

> Zed v1.22.0 lets subagents select their own model via `spawn_agent`'s new `model` parameter and adds automatic subagent compaction, adds BYOK support for GPT-6.1 Sol and Grok 4.7, extends several config keys with a `"..."` inheritance placeholder, and ships two breaking changes around Threads Sidebar settings and external-agent terminal controls.

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

01 Config inheritance via "..." spread syntax IMPROVED

85

Several Zed settings now support a `"..."` placeholder to extend inherited values instead of overriding them wholesale: `edit_predictions.disabled_globs` and `edit_predictions_disabled_in` for edit-prediction exclusions, `terminal.path_hyperlink_regexes` for path hyperlink regexes, and `file_scan_inclusions` and `hidden_files` so project settings can extend inherited inclusion and hidden-file patterns.

Extend your project's inherited file-scan inclusions with additional patterns without repeating the defaults.

```json
{ "file_scan_inclusions": ["...", "vendor/**", "*.generated.*"]
}
```

— Names every extended config key, with a working example. v1.22.0

02 Threads Sidebar auto-open setting NEW

80

A new `agent.threads_sidebar.auto_open` setting controls whether opening a folder in an existing window also opens the Threads Sidebar; set it to `false` to disable this.

Prevent the Threads Sidebar from auto-opening every time you open a folder in an existing Zed window.

```json
{ "agent": { "threads_sidebar": { "auto_open": false } } }
```

— Exact config key and working example provided. v1.22.0

03 Unified/split toggle in diff view NEW

70

The diff view opened with `zed --diff <old> <new>` now includes a toggle to switch between unified and split display modes.

— Exact command given; toggle mechanism not detailed. v1.22.0

04 **Subagent model selection and automatic compaction** NEW

65

The `spawn_agent` tool now accepts an optional `model` parameter to choose a model for that specific subagent spawn instead of inheriting the configured or parent model, and Zed now enables automatic compaction for subagents.

— Names the parameter but gives no usage example. v1.22.0

05 **Threads Sidebar settings renamed** BREAKING

65

Threads Sidebar settings are renamed to `agent.threads_sidebar.position`, `agent.threads_sidebar.default_width`, and related keys; existing config using the old names will need updating.

— Names exact replacement keys and migration need. v1.22.0

THINNER COVERAGE BELOW

06 **BYOK GPT-6.1 Sol and Grok 4.7 model support** NEW

55

Adds BYOK (bring-your-own-key) support for GPT-6.1 Sol using an OpenAI API key, and adds Grok 4.7 as the recommended default model for the SuperGrok and xAI providers.

— Names models and providers but no setup steps. v1.22.0

07 **Dynamic OpenCode model list fetching** IMPROVED

50

Improves OpenCode model availability by fetching the Zen and Go model list dynamically instead of relying on a bundled list.

— Explains the change but no user-facing action. v1.22.0

08 **Mica backdrop options for Windows theme** NEW

50

Adds `MicaBackdrop` and `MicaAltBackdrop` options for the `background.appearance` theme setting on Windows.

— Named options but no usage example given. v1.22.0

09 **Stop controls removed from external-agent terminals** BREAKING

45

Stop controls are removed from external-agent-managed terminals, which are now controlled by the agent rather than a local process.

— States the change but gives no migration step. v1.22.0

10 **Three-finger gestures in Markdown previews on macOS** NEW

40

Adds macOS three-finger Look Up and translation gestures for text in focused Markdown previews.

— Describes the gesture with no further mechanism. v1.22.0

└──▷ BREAKING ON UPGRADE

! Stop controls are removed from external-agent-managed terminals, which are now controlled by the agent rather than a local process.

! Threads Sidebar settings are renamed to `agent.threads_sidebar.position`, `agent.threads_sidebar.default_width`, and related keys — existing config using the old names will need updating.

v1.22.0 NOTES ↗ LATEST STABLE

Zed v1.22.0 adds BYOK GPT-6.1 Sol, per-subagent model selection, subagent compaction, and new config keys for sidebar, diffs, and path patterns.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.22.0 https://github.com/zed-industries/zed.git
# already have the repo? check out this version:
$ git checkout v1.22.0
```

└──▷ USE IT

Prevent the Threads Sidebar from auto-opening every time you open a folder in an existing Zed window.

~/.config/zed/settings.json

```
{ "agent": { "threads_sidebar": { "auto_open": false } } }
```

Extend your project's inherited file-scan inclusions with additional patterns without repeating the defaults.

.zed/settings.json

```
{ "file_scan_inclusions": ["...", "vendor/**", "*.generated.*"] }
```

Adds optional `model` parameter to `spawn_agent` to choose a model for that subagent spawn instead of inheriting the configured or parent model.

Adds `agent.threads_sidebar.auto_open` setting to control whether opening a folder

in an existing window also opens the Threads Sidebar.

Adds `MicaBackdrop` and `MicaAltBackdrop` options for the `background.appearance` theme setting on Windows.

Adds support for `"..."` in `edit_predictions.disabled_globs` and `edit_predictions_disabled_in` to extend inherited edit-prediction exclusions without repeating them.

Adds support for `"..."` in `terminal.path_hyperlink_regexes` to extend inherited path hyperlink regexes without repeating them.

+7 more

└─▷ BREAKING ON UPGRADE

- ! Stop controls are removed from external-agent-managed terminals, which are now controlled by the agent rather than a local process.
- ! Threads Sidebar settings are renamed to `agent.threads_sidebar.position`, `agent.threads_sidebar.default_width`, and related keys — existing config using the old names will need updating.

WAS THIS USEFUL? 👍 👎

Amazon Kiro

Feedback ⓘ

Web: Introducing Workflows in Kiro Web: Run reusable, multi-step agent plans in Kiro W

BLOG ↗ RANK

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

> Kiro's biggest release of the window is Workflows — reusable, multi-step agent plans that now run across Kiro Web, CLI, and IDE — alongside tighter approval safeguards for untrusted workspaces, a Classic session deprecation notice, and new enterprise sign-in and telemetry controls.

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

01 **Workflows across Web, CLI, and IDE** NEW 97

Kiro ships Workflows — reusable, multi-step agent plans — simultaneously across its three surfaces. In Kiro Web, enable via `Open Settings > Workflows` and toggling Enable workflows on; upload a `.workflow.json`, `.workflow.yaml`, or `.workflow.yml` recipe, or import a `.kiro/workflows/` folder via Configuration Sync. Three bundled recipes ship: `investigate` (read-only codebase investigation), `feature-pipeline` (staged feature delivery with independent reviewers), and `publish-pr` (open a PR and handle review feedback); a Workflows side panel shows each step's status, elapsed time, agent, model, effort, and tool activity, with interactive controls Answer this step, Steer the agent, pause, resume, stop, and retry; each step runs in its own isolated agent session receiving only outputs handed off by earlier steps, pausing automatically when human input is needed. In the CLI (V3), `/workflow run` and `/workflow` commands launch, inspect, and manage runs, gated by an opt-in toggle at `/settings` > Features > Workflows (requires restart). In the IDE (1.2), the `kiroAgent.workflows.enabled` setting (or the Workspace Configuration > Agent Focus toggle) turns it on, recipes live in `.kiro/workflows/` and are not loaded in an untrusted workspace until trusted, and dedicated Workflows and Artifacts views in Agent Focus Mode show step activity, artifact previews, and an Open in IDE action on every artifact.

Enable Workflows in Kiro Web so you can run multi-step agent plans in background cloud sessions.

📍 1. Open Settings › Workflows 2. Toggle Enable workflows on 3. Click Upload workflow and select your .workflow.json, .workflow.yaml, or .workflow.yml recipe file — or use Configuration Sync to import a full .kiro/workflows/ folder

Use a bundled recipe to open a pull request and handle review feedback automatically, then monitor each step as it runs.

📍 1. In the parent conversation, describe the outcome and select the `publish-pr` recipe when Kiro proposes the Workflow 2. Once the run starts, open `Workflows` from the side panel and select the active run 3. Select any step to view its status, elapsed time, agent, model, effort, and tool activity 4. When a step pauses for input, click `Answer this step` or `Steer the agent` to guide it forward

Enable the Workflows feature in a V3 session so you can run reusable multi-step tasks without re-describing them each time.

📍 In the Kiro CLI V3 session, type `/settings`, navigate to Features, and enable Workflows, then restart Kiro CLI to apply the change.

Launch a pre-defined Workflow recipe by name to automate a repeatable multi-step task from the CLI.

```
$ /workflow run
```

Enable Workflows for a project so Kiro can run multi-step agent plans in the background while you keep working in the parent chat.

📍 In Settings, search for `kiroAgent.workflows.enabled` and set it to true, then start a new chat session. Use the Workflows area above the chat to monitor step progress and pause, resume, or stop the run.

— Names every flag, command, and config path across three surfaces

Web: Introducing Workflows in Kiro Web:… | CLI: Workflows, Classic Migration Guida… | IDE: Workflows, Safer Untrusted Workspa…

---

02 **Stricter V3 approval checks for files, tools, shells** BREAKING

64

V3 Hook and LSP file writes now require per-file approval before writing a Hook file and before each file in a rename or format operation. V3 MCP tools in untrusted workspaces now request approval on every call and retry, confirming the approval is still current immediately before each call runs. V3 shell commands in untrusted workspaces now request approval on each start even when a saved or broad allowance exists, and V3 tool execution rechecks each approval immediately before the action runs so an approval changed or cancelled while waiting is not applied.

— Describes changed approval behavior but gives no command to test it

CLI: Workflows, Classic Migration Guida…

THINNER COVERAGE BELOW

03 **Untrusted workspace safeguards for Kiro configuration** BREAKING

58

Kiro now prompts before the agent modifies agent, Hook, or Power files, or any file in `.kiro/workflows/`, even when broader file permissions would allow the write. In an untrusted workspace, Kiro also prompts before changing any other Kiro configuration and before each command runs, overriding previously granted permissions.

— Explains the override behavior but no setting to toggle it

IDE: Workflows, Safer Untrusted Workspa…

04 **Enterprise sign-in controls** NEW

44

Adds enterprise sign-in controls so administrators can restrict permitted sign-in methods, prefill organization sign-in details, and add a help link to the sign-in screen.

— Lists admin capabilities but no config key or path

IDE: Workflows, Safer Untrusted Workspa…

05 **Classic session deprecation notice** DEPRECATED

40

Classic sessions now display a deprecation notice that identifies the specific flag, environment variable, or saved setting that launched Classic and points to switching guidance.

— States the notice exists but not the switching steps

CLI: Workflows, Classic Migration Guida…

06 **Regional telemetry routing for enterprise profiles** IMPROVED

37

Kiro now routes telemetry and activity data to the AWS Region of the selected enterprise profile, keeping enterprise traffic regional.

— States the behavior with no verification step or setting

IDE: Workflows, Safer Untrusted Workspa…

Web: Introducing Workflows in Kiro Web: Run reusable, multi-step agent plans in Kiro

BLOG ↗ LATEST STABLE

Kiro Web gains Workflows: reusable, multi-step agent plans that run in background cloud sessions while you keep working.

└──▷ HOW TO FIND IT

Enable Workflows in Kiro Web so you can run multi-step agent plans in background cloud sessions.

📍 1. Open Settings › Workflows 2. Toggle Enable workflows on 3. Click Upload workflow and select your .workflow.json, .workflow.yaml, or .workflow.yml recipe file

— or use `Configuration Sync` to import a full .kiro/workflows/ folder

Use a bundled recipe to open a pull request and handle review feedback automatically, then monitor each step as it runs.

📍 1. In the parent conversation, describe the outcome and select the `publish-pr` recipe when Kiro proposes the Workflow 2. Once the run starts, open `Workflows` from the side panel and select the active run 3. Select any step to view its status, elapsed time, agent, model, effort, and tool activity 4. When a step pauses for input, click `Answer this step` or `Steer the agent` to guide it forward

Enables Workflows via `Open Settings > Workflows` and toggling `Enable workflows` on, hiding all Workflow surfaces when turned off.

Introduces three bundled workflow recipes: `investigate` (read-only codebase investigation), `feature-pipeline` (staged feature delivery with independent reviewers), and `publish-pr` (open a pull request and handle review feedback).

Adds a Workflows side panel in the parent conversation to inspect nested steps, view status, elapsed time, agent, model, effort, and tool activity for each step.

Supports interactive step controls — `Answer this step`, `Steer the agent`, pause, resume, stop, and retry failed steps — from within the Workflows panel.

Runs each workflow step in its own isolated agent session that receives only outputs handed off by earlier steps, pausing automatically when human input is required.

`CLI: Workflows, Classic Migration Guidance, and V3 Approval Safeguards: Run reusable`

BLOG ↗ STABLE

Kiro CLI 2.26.0 adds opt-in Workflows for reusable multi-step tasks, Classic migration guidance, and tighter V3 approval safeguards.

└──▷ HOW TO FIND IT

Enable the Workflows feature in a V3 session so you can run reusable multi-step tasks without re-describing them each time.

📍 In the Kiro CLI V3 session, type `/settings`, navigate to Features, and enable Workflows, then restart Kiro CLI to apply the change.

Launch a pre-defined Workflow recipe by name to automate a repeatable multi-step task from the CLI.

```
$ /workflow run
```

Adds `/workflow run` and `/workflow` management commands in V3 to launch, inspect, and manage reusable multi-step Workflow runs.

Adds opt-in Workflows feature enabled via `/settings` > Features > Workflows in V3 (requires restart to apply).

Classic sessions now display a deprecation notice that identifies the specific flag, environment variable, or saved setting that launched Classic and points to switching guidance.

V3 Hook and LSP file writes now require per-file approval before writing a Hook file and before each file in a rename or format operation.

V3 MCP tools in untrusted workspaces now request approval on every call and retry, and confirm approval is still current immediately before each call runs.

+2 more

`IDE: Workflows, Safer Untrusted Workspaces, and Enterprise Sign-In Controls: Run reu`

BLOG ↗ STABLE

Kiro 1.2 adds reusable multi-agent Workflows, stronger untrusted-workspace safeguards, and enterprise sign-in controls.

└──▷ HOW TO FIND IT

Enable Workflows for a project so Kiro can run multi-step agent plans in the background while you keep working in the parent chat.

> 📍 In Settings, search for `kiroAgent.workflows.enabled` and set it to true, then start a new chat session. Use the Workflows area above the chat to monitor step progress and pause, resume, or stop the run.

Adds `kiroAgent.workflows.enabled` setting (or Workspace Configuration > Agent Focus toggle) to turn on background multi-agent Workflows, where each step runs in its own agent session and can be paused, resumed, or stopped.

Workflow recipes stored in `.kiro/workflows/` define reusable multi-step plans; they are not loaded in an untrusted workspace until the workspace is trusted.

Kiro now prompts before the agent modifies agent, Hook, or Power files, or any file in `.kiro/workflows/`, even when broader file permissions would allow the write.

In an untrusted workspace, Kiro also prompts before changing any other Kiro configuration and before each command runs, overriding previously granted permissions.

Adds dedicated Workflows and Artifacts views in Agent Focus Mode session details, including step activity, artifact previews, and an `Open in IDE` action on every artifact.

+2 more

WAS THIS USEFUL? 👍 👎

Superset

Feedback ⓘ cli-v1.33.0 NOTES ↗ STABLE RANK

Superset is an agentic IDE to orchestrate 100+ coding agents in parallel. Run any agent with your own subscription.

CLI gains `workspaces get` and configurable `set-wake` / `wake` commands; desktop adds Mistral agent, host deletion, and terminal diagnostics.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch cli-v1.33.0 https://github.com/superset-sh/superset.git
# already have the repo? check out this version:
$ git checkout cli-v1.33.0
```

└──▷ TRY IT

Retrieve a specific workspace by its ID from the CLI, useful in scripts that need to verify a workspace exists before connecting.

```
$ superset workspaces get <workspace-id>
```

Trigger the configured wake command for an offline host before launching a remote session.

```
$ superset wake <host-id>
```

Adds `workspaces get` CLI subcommand to look up a workspace by ID.

Adds configurable per-host `set-wake` and `wake` CLI commands to control host wake behaviour.

Adds first-class Mistral Vibe coding agent to the desktop.

Adds Sydney (`syd`) region to the relay fleet, expanding geographic coverage.

Enables live chat sessions routed over the relay on mobile.

+18 more

NEW INTEGRATION Sentry plugin via sentry_mcp connector

▾ 1 more release in this issue · 2026-09-30

desktop-v1.33.0 NOTES ↗ LATEST STABLE

v1.33.0 upgrades cloud workspace collaboration, adds Claude Sonnet 5.5 to model pickers, and ships UI navigation improvements across desktop and mobile.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch desktop-v1.33.0 https://github.com/superset-sh/superset.git
# already have the repo? check out this version:
$ git checkout desktop-v1.33.0
```

Cloud workspaces and tasks become shared records with presence, sharing, descriptions, comments, labels, and mentions.

Offers Claude Sonnet 5.5 as a selectable option in model pickers.

Enables inviting members directly from every people picker in the desktop app.

Pins Claude and Codex in the sandbox bundle so boxes update on wake.

Moves ship controls into the workspace sidebar strip.

+2 more

WAS THIS USEFUL? 👍 👎

Diagram Design

Feedback ⓘ changes since 2026-09-01 CODE ↗

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

Adds waterfall and heatmap chart types, Marimekko treemap variant, Streamgraph line variant, Excalidraw import, and two new semantic patterns.

Adds `print-a3-landscape` size preset to the output spec for A3 landscape print output.

Adds waterfall chart as a new visual type.

Adds heatmap as visual type #40.

Supports importing Excalidraw diagrams for editorial redraw.

Adds traceable block decomposition semantic pattern.

+3 more

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

Feynman

Feedback ⓘ v0.5.19 NOTES ↗ LATEST STABLE RANK

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

Feynman v0.5.19: LibreOffice-free .docx reading, workflow scripts now written to outputs/.plans/, and subagent trust/reload fixes.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.5.19 https://github.com/Companion-Inc/feynman.git
# already have the repo? check out this version:
$ git checkout v0.5.19
```

Adds fallback `.docx` parsing to `document_parse`: when LibreOffice is absent, reads paragraph text directly from the file and surfaces a notice; install LibreOffice for full fidelity (page layout, tables, footnotes, images).

`feynman_science_database_search` now accepts `"null"` for the `sort` and `limit` parameters, unblocking searches where models omit those fields.

`/deepresearch` and `/summarize` now write their workflow script to `outputs/.plans/<slug>-workflow.js` and execute it by path, replacing the removed `workflowScript` parameter.

Subagent children now inherit the project's trust setting.

UPDATED INTEGRATION alphaXiv fallback to OpenAlex and arXiv

UPDATED INTEGRATION Semantic Scholar / OpenAlex mutual fallback for paper searches

UPDATED INTEGRATION OpenAI web search via ChatGPT login (pi-web-access 0.35.0)

└──▷ BREAKING ON UPGRADE

! The `workflowScript` parameter has been removed from pi-subagents; `/deepresearch` and `/summarize` now write the workflow script to `outputs/.plans/<slug>-workflow.js` and run it by path instead.

▾ 2 more releases in this issue · 2026-09-30 → 2026-10-01

v0.5.18 NOTES ↗ STABLE

Feynman v0.5.18 adds automatic resilience across research index sources and restores web search via ChatGPT login.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.5.18 https://github.com/Companion-Inc/feynman.git
# already have the repo? check out this version:
$ git checkout v0.5.18
```

Adds automatic cross-index fallback between Semantic Scholar and OpenAlex for paper searches, with each result indicating which index answered and why.

Adds fallback for `alpha_search` and `alpha_get_paper` when alphaXiv is unreachable; `feynman alpha login` restores full alphaXiv responses.

Adds OpenAI web search support when authenticated via ChatGPT using `feynman model login openai`.

v0.5.16 NOTES ↗ STABLE

Feynman v0.5.16 adds `feynman packages remove` and fixes subagent output paths to resolve against the workspace.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.5.16 https://github.com/Companion-Inc/feynman.git
# already have the repo? check out this version:
$ git checkout v0.5.16
```

└──▷ TRY IT

Uninstall an optional package (e.g. `memory`) you no longer want loaded into every session.

```
$ feynman packages remove memory
```

Pin a custom subagent artifact root so your CI picks up files from a known path rather than the workspace default.

~/.feynman/agent/extensions/subagent/config.json

```
{"singleRunOutputBaseDir": "/data/feynman-artifacts"}
```

Adds `feynman packages remove <preset>` (aliases: `rm`, `uninstall`) to remove optional packages such as `memory`.

> `/deepresearch` and `/lit` now name every subagent file under `outputs/.drafts/`, and the researcher, reviewer, verifier, and writer no longer default to generic file names such as `research.md`.

WAS THIS USEFUL? 👍 👎

LibreChat

Feedback ⓘ v0.8.8 NOTES ↗ RANK

LibreChat provides a self-hosted chat interface with multiple model providers, agents, tools, and retrieval over uploaded files.

> LibreChat v0.8.8 ships a broad agent-platform expansion — a Beta Agents API, agent interrupt/steer/approve controls, Code Workspaces, Scheduled Chats, and Execution Visibility — alongside new model support and smaller chat UI and routing improvements.

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

01 **Execution Visibility for agent runs** NEW 70

Adds Execution Visibility with live activity, tool timing, Context Usage, optional costs, and a configured Langfuse Trace Viewer, distinguishing tool preparation time from tool call time, plus a streaming thought peek under collapsed live cards.

— Mechanism detailed, no exact config key v0.8.8

02 **Scheduled Chats (beta)** NEW 65

Introduces Scheduled Chats (beta, disabled by default) with cron scheduling, time zone support, and Project destinations.

— Explains scope and config but no exact flag or UI path v0.8.8

03 **Beta Agents API** NEW 60

Introduces a Beta Agents API with improved Chat Completions and Open Responses compatibility, Agent/Skill management, and Agent Events.

— Names API surfaces but no endpoint paths given v0.8.8

04 **Code Workspaces for agent sessions** NEW 60

Introduces Code Workspaces with stateful sessions, attached machines, workspace recovery, and guarded permissions (highly experimental). Users can attach or leave a workspace on a saved chat.

— Describes capabilities, no exact commands v0.8.8

05 **Background Tasks with task cards and header controls** NEW 60

Adds Background Tasks with task cards, header controls, and automatic delivery of supported results, rendered as native tool cards in the chat UI. Users can view and stop background tasks directly from the chat header.

— Describes UI surfaces, no exact commands v0.8.8

THINNER COVERAGE BELOW

06 **Unified Agent Builder with Skill authoring** NEW 50

Adds Unified Agent Builder with same-run Skill authoring, opt-in Programmatic Tool Calling, and background tools.

— Names features but no concrete steps given v0.8.8

07 **Claude Sonnet 5.5 and GPT-6.1 Sol model support** NEW 50

Adds Claude Sonnet 5.5 and GPT-6.1 Sol model support, with GPT-6.1 Sol using family-fallback limits and pricing.

— Names models but no setup steps v0.8.8

08 **Agent interrupt, steer, and approval controls** NEW 45

Adds agent control: `interrupt`, `steer`, queue messages, answer questions, and approve tool calls.

— Lists controls but no mechanism or UI path v0.8.8

09 **Experimental Agent Plugins bundling Skills and MCP** NEW 45

Adds Experimental Agent Plugins that bundle Skills and MCP servers with opt-in command hooks.

— Thin description of plugin mechanism v0.8.8

10 **Tenant-scoped YAML custom endpoints** NEW 45

YAML custom endpoints can now be scoped by tenant, enabling tenant-isolated operator configuration.

— No config key name given for tenant scoping v0.8.8

11 **Per-worktree request routing** NEW 35

Adds per-worktree request routing by routing linked worktree requests into per-worktree lanes.

— Mechanism named but no config or flag given v0.8.8

12 **Bounded PII text transformation groundwork** NEW 35

Adds Bounded PII Text Transformation groundwork (library only; message ingress blocks matches).

— Library-only groundwork, not yet user-facing v0.8.8

13 **Chat filtering and draggable pins** NEW 30

Adds chat filtering and draggable pins to the chat UI.

— Bare description with no further detail v0.8.8

14 **Manual context compaction for long threads** NEW 30

Adds manual compaction for long-thread context management.

— No mechanism or trigger specified v0.8.8

v0.8.8 NOTES ↗ STABLE

LibreChat v0.8.8 adds agent interrupt/steer controls, scheduled chats, code workspaces, a Beta Agents API, and tenant-scoped YAML endpoints.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.8.8 https://github.com/LibreChat-AI/LibreChat.git
# already have the repo? check out this version:
$ git checkout v0.8.8
```

Scopes YAML custom endpoints by tenant, enabling tenant-isolated operator configuration.

Adds Claude Sonnet 5.5 and GPT-6.1 Sol model support, with GPT-6.1 Sol using family-fallback limits and pricing.

Introduces Scheduled Chats (beta, disabled by default) with cron scheduling, time zone support, and Project destinations.

Introduces Code Workspaces with stateful sessions, attached machines, workspace recovery, and guarded permissions (highly experimental).

Introduces a Beta Agents API with improved Chat Completions and Open Responses compatibility, Agent/Skill management, and Agent Events.

+14 more

WAS THIS USEFUL? 👍 👎

LangChain Deep Agents

Feedback ⓘ deepagents==0.7.21

NOTES ↗ RANK

Deep Agents provides an agent harness with planning, filesystem tools, subagents, and context management.

> Deep Agents 0.7.21 extends its FilesystemMiddleware to support storing blobs on the backend rather than only locally.

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

01 Backend blob storage in FilesystemMiddleware IMPROVED

40

`FilesystemMiddleware` can now store blobs on the backend, extending the filesystem middleware beyond local file handling.

— Names the component but no config details or mechanism deepagents==0.7.21

deepagents==0.7.21 NOTES ↗ LATEST STABLE

FilesystemMiddleware gains backend blob storage support in deepagents 0.7.21

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch deepagents==0.7.21
https://github.com/langchain-ai/deepagents.git
# already have the repo? check out this version:
$ git checkout deepagents==0.7.21
```

Enables `FilesystemMiddleware` to store blobs on the backend, extending filesystem middleware beyond local file handling.

WAS THIS USEFUL? 👍 👎

OpenWorker

Feedback ⓘ v0.3.0 CODE ↗ RANK

OpenWorker automates desktop tasks across files, tools, and integrations with governed, audited AI agent actions.

> OpenWorker v0.3.0 introduces a full remote-machine management system with CLI, auth and GUI surfaces, an MCP permission model, a Coworkers subsystem, GUI internationalization, and an expanded model matrix, alongside several reply-generation controls.

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

01 **Remote machine management system** NEW 95

Adds `openworker machine` CLI subcommands — `openworker join`, `openworker up`, `machine logs`, `machine status --json` — to register, connect, and manage remote machines, backed by a launchd service on macOS and an `install.sh` script. Remote boxes persist connection settings in a `remote.json` config file (keyed by `CONFIG_FILENAME`) in the state dir, dial the controller, and authenticate via an ed25519 challenge-response or a 10-minute enrollment token, exposing themselves through the `/ws/machine` WebSocket endpoint and the `/v1/machines/{mid}/p/<path>` proxy prefix. The GUI gains a sidebar-by-machine app shell with a runs-on chip, composer, transcript view, a Settings > Machines scoped page, and per-machine connectors for GitHub and Slack.

Check the live status of a registered remote machine in JSON for scripting or monitoring pipelines.

```
$ openworker machine status --json
```

Stream recent log output from a remote machine to debug a failing run.

```
$ openworker machine logs <machine-id>
```

— Full CLI, auth mechanism, and endpoints named verbatim v0.3.0

02 **Per-reply output-token limit with truncation continuation** NEW 75

Adds a configurable per-reply output-token limit (OPE-177) so callers can cap how many tokens a single assistant reply may consume. When a reply is cut off at that limit with no action taken, OpenWorker automatically continues the turn, retrying up to

`MAX_TRUNCATION_CONTINUATIONS` times before ending the turn as `truncated` (OPE-171).

— Names the limit, retry constant, and terminal state v0.3.0

03 **MCP permission model with trust and approvals** NEW 60

Adds an MCP permission model with an EXTERNAL floor, durable per-tool trust, run grants, and honest approval cards in the GUI (OPE-136).

— Names mechanism but no config surface to act on v0.3.0

THINNER COVERAGE BELOW

04 **Coworkers subsystem with personas and team store** NEW 55

Adds a Coworkers subsystem with per-coworker model selection, reviewer and lead personas, and a team store, including engine-side worker grants, lead decisions, and parked gates.

— Describes components but no concrete commands v0.3.0

05 **Configurable reasoning effort setting** NEW 55

Adds a `reasoning_effort` setting, translated per provider and recorded per reply, to control model reasoning depth.

— Names the setting but not its accepted values v0.3.0

06 **Model matrix adds Claude 5 and Kimi K3** NEW 50

Adds Claude 5, Kimi K3 (via OpenRouter, 1,048,576-context) to the model matrix, and refreshes context-window data for Kimi, GLM, and DeepSeek.

— Names models and context sizes but no selection flag v0.3.0

07 **Persisted finish_reason and served_by on replies** NEW 45

Persists `finish_reason` on assistant messages and `served_by` — the upstream host a router reports — on each reply.

— Names fields but not how to query them v0.3.0

08 **GUI internationalization with English and Chinese locales** NEW 40

Adds GUI internationalization (i18n) infrastructure with English and Chinese locales, covering new screens for machines, connectors, teams, and settings.

— States locales and screens but no toggle mechanism v0.3.0

09 **Dev-only card gallery for debugging** NEW 40

Adds a card gallery (dev-only) for inspecting card states, scenario sessions, and server contract checks.

— Names purpose but no access path given v0.3.0

10 **Reasoning token replay on OpenAI-compatible path** IMPROVED

30

Replays reasoning tokens on the OpenAI-compatible provider path (OPE-178).

— One-line fix with no behavioural detail v0.3.0

v0.3.0 CODE ↗ LATEST STABLE

OpenWorker v0.3.0 adds remote machine management, an MCP permission model, multi-lingual UI, and configurable reasoning effort.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.3.0
https://github.com/andrewyng/openworker.git
# already have the repo? check out this version:
$ git checkout v0.3.0
```

└──▷ TRY IT

Check the live status of a registered remote machine in JSON for scripting or monitoring pipelines.

```
$ openworker machine status --json
```

Stream recent log output from a remote machine to debug a failing run.

```
$ openworker machine logs <machine-id>
```

Adds `openworker machine` CLI subcommands (including `openworker join`, `openworker up`, `machine logs`, `machine status --json`) to register, connect, and manage remote machines from the command line, with a launchd service on macOS and an `install.sh` script.

Adds `remote.json` config file (stored in the state dir, keyed by `CONFIG_FILENAME`) to persist remote-machine connection settings on the box side.

Adds a `reasoning_effort` setting, translated per provider and recorded per reply, to control model reasoning depth.

> Adds a configurable per-reply output-token limit (OPE-177) so callers can cap how many tokens a single assistant reply may consume.
>
> Introduces a remote-machine architecture: remote boxes dial the controller via `openworker join` / `openworker up`, authenticate with an ed25519 challenge-response or a 10-minute enrollment token, and expose themselves through the `/ws/machine` WebSocket endpoint and the `/v1/machines/{mid}/p/<path>` proxy prefix.
>
> +9 more

WAS THIS USEFUL? 👍 👎

LangChain

Feedback ⓘ langchain-openai==1.6.7 NOTES ↗ RANK

LangChain is an open-source framework that orchestrates applications powered by language models.

> langchain-openai 1.6.7 adds automatic discovery of Azure workload identity credentials, enabling passwordless authentication to Azure OpenAI.

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

01 **Automatic Azure workload identity discovery** NEW 45

langchain-openai now automatically discovers Azure workload identity credentials, enabling passwordless authentication to Azure OpenAI without explicit credential configuration.

— Describes mechanism and benefit but no config keys or code shown

langchain-openai==1.6.7

langchain-openai==1.6.7 NOTES ↗ LATEST STABLE

langchain-openai 1.6.7 adds automatic Azure workload identity discovery for passwordless auth.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch langchain-openai==1.6.7
https://github.com/langchain-ai/langchain.git
# already have the repo? check out this version:
$ git checkout langchain-openai==1.6.7
```

WAS THIS USEFUL? 👍 👎

Microsoft Agent Framework

Feedback ⓘ dotnet-1.23.0

NOTES ↗ RANK

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

> Microsoft Agent Framework's dotnet-1.23.0 release adds origin pinning to the Foundry MCP client and middleware tool-substitution support, alongside a cluster of breaking changes around dependency versions, approval workflows, run-state tool handling, and configuration key validation.

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

01 **Dependency version bumps across SDKs** BREAKING 70

Bumps `Azure.AI.Projects` to `3.0.0-beta.3`, `OpenAI` to `2.14.0`, and `MEAI` to `10.10.1`, and updates AGUI SDK packages to `1.0.0`; dependent code using removed or changed APIs in these libraries will break on upgrade.

— Names exact package versions upgraders must reconcile. dotnet-1.23.0

02 **GitHub Copilot SDK session integration** IMPROVED 65

Bumps `GitHub.Copilot.SDK` to `1.0.1` and forwards session config properties including per-session `GitHubToken`.

— Names the exact SDK version and config field for session auth. dotnet-1.23.0

THINNER COVERAGE BELOW

03 **Approval workflow behavior changes** BREAKING 50

Approval response binding is now enforced consistently (#8641), so workflows relying on previously permissive binding behaviour will break, and DevUI approval continuation is changed (#8423), meaning in-flight DevUI approval flows may not resume correctly after upgrade.

— Names both PRs and effects but no remediation steps. dotnet-1.23.0

04 **Tool-change handling between runs** BREAKING 45

Better support for tool changes between runs (#8754) — existing serialized run state referencing prior tool definitions may be incompatible with this change.

— Flags the incompatibility risk but no migration steps given. dotnet-1.23.0

05 **Allow list for configuration keys** BREAKING 45

Configuration keys now use an allow list (#8200); any config keys not on the allow list will no longer be accepted.

— Names the behaviour change but not the allow list contents. dotnet-1.23.0

06 **Origin pinning for Foundry MCP client** NEW 35

Adds origin pinning to the Foundry toolbox MCP client for supply-chain integrity.

— Describes the safeguard but no config flag or verification step named. dotnet-1.23.0

07 **Function replace support in middleware** NEW 35

Adds function replace support for function middleware, enabling middleware to substitute tool implementations at runtime.

— Explains the mechanism but gives no API or code example. dotnet-1.23.0

08 **McpDocsResearch declarative workflow sample** NEW 35

Adds a `McpDocsResearch` declarative workflow sample showcasing the agent-level MCP pattern.

— Names the sample but gives no code or usage walkthrough. dotnet-1.23.0

09 **Cancellation token propagation in TextSearchProvider** IMPROVED 35

Propagates `TextSearchProvider` caller cancellation tokens through the search pipeline.

— Names the component but not how to invoke cancellation. dotnet-1.23.0

10 **Durable storage for external input messages** IMPROVED 30

Stores created external input messages for durability across workflow runs.

— States the behaviour change without a named key or API. dotnet-1.23.0

╰──▷ BREAKING ON UPGRADE

! Better support for tool changes between runs (#8754) — existing serialized run state referencing prior tool definitions may be incompatible.

! `Azure.AI.Projects` bumped to `3.0.0-beta.3`, `OpenAI` to `2.14.0`, and `MEAI` to `10.10.1` — dependent code using removed or changed APIs in these libraries will break on upgrade.

! Configuration keys now use an allow list (`fix: use allow list for configuration keys`, #8200) — any config keys not on the allow list will no longer be accepted.

! Approval response binding is now enforced consistently (#8641) — workflows relying on previously permissive binding behaviour will break.

! DevUI approval continuation is changed (#8423) — in-flight DevUI approval flows may not resume correctly after upgrade.

`dotnet-1.23.0` NOTES ↗ LATEST STABLE

dotnet-1.23.0 adds origin pinning to Foundry MCP client, function replace support for middleware, and bumps `Azure.AI.Projects` to 3.0.0-beta.3.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch dotnet-1.23.0
https://github.com/microsoft/agent-framework.git
# already have the repo? check out this version:
$ git checkout dotnet-1.23.0
```

Adds origin pinning to the Foundry toolbox MCP client for supply-chain integrity.

Adds function replace support for function middleware, enabling middleware to substitute tool implementations at runtime.

Bumps `Azure.AI.Projects` to `3.0.0-beta.3`, `OpenAI` to `2.14.0`, and `MEAI` to `10.10.1`.

Updates AGUI SDK packages to `1.0.0`.

Adds a `McpDocsResearch` declarative workflow sample showcasing the agent-level MCP pattern.

+3 more

└──▷ BREAKING ON UPGRADE

! Better support for tool changes between runs (#8754) — existing serialized run state referencing prior tool definitions may be incompatible.

! `Azure.AI.Projects` bumped to `3.0.0-beta.3`, `OpenAI` to `2.14.0`, and `MEAI` to `10.10.1` — dependent code using removed or changed APIs in these libraries will break on upgrade.

! Configuration keys now use an allow list (`fix: use allow list for configuration keys`, #8200) — any config keys not on the allow list will no longer be accepted.

! Approval response binding is now enforced consistently (#8641) — workflows relying on previously permissive binding behaviour will break.

! DevUI approval continuation is changed (#8423) — in-flight DevUI approval flows may not resume correctly after upgrade.

WAS THIS USEFUL? 👍 👎

Agno (formerly Phidata)

Feedback ⓘ v3.1.0 NOTES ↗

RANK

Agno is an agent framework and runtime that orchestrates and runs multi-agent systems.

> Agno v3.1 adds role-based access control and a database-backed filesystem to AgentOS, ships a new AI/ML API toolkit, and flips MCP tool exposure to opt-in with a mandatory filesystem migration for existing deployments.

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

01 **Database-backed filesystem for AgentOS** BREAKING 95

Adds `agno.fs.DbFileSystem` backed by an `agno_fs` table with `/filesystem` routes for listing, reading, and managing files. The table is now keyed by (namespace, user_id, path); upgrading from an earlier release raises `SchemaOutdatedError` until migrated manually — run `python libs/agno/migrations/migrate_filesystem_postgres.py` (PostgreSQL) or `python libs/agno/migrations/migrate_filesystem_sqlite.py` (SQLite) with the application stopped, since the migration never runs automatically.

— Exact table, routes, and migration commands given v3.1.0

02 **MCP default and lifecycle tools now opt-in** BREAKING 90

MCPConfig(tools=[...]) now publishes exactly the tools listed — `default_tools` and `lifecycle_tools` default to `False`, so built-in default tools and lifecycle tools (`continue_run` / `cancel_run`) are no longer served automatically. Agno also refreshed MCP server config and built-in MCP auth handling. To restore prior behavior: MCPConfig(tools=["my_custom_tool"], default_tools=True, lifecycle_tools=True).

Opt in to MCP lifecycle tools explicitly now that `default_tools` and `lifecycle_tools` are `False` by default in `MCPConfig`.

python

```python
from agno.mcp import MCPConfig

mcp = MCPConfig(
    tools=["my_custom_tool"],
    default_tools=True,
    lifecycle_tools=True,
)
```

— Exact flags and a runnable migration snippet provided v3.1.0

03 **RBAC for AgentOS** NEW 73

Adds the `agno.os.authz` package with role-based access control (RBAC) for AgentOS — includes a role store, scope policy, audit log, user directory, admin router, and pluggable authorization engines (`native` and fine-grained `fga`).

— Lists components and engines but no setup command shown v3.1.0

THINNER COVERAGE BELOW

04 **AIMLAPITools toolkit** NEW 44

Adds the AIMLAPITools toolkit for image, video, speech, and transcription via the AI/ML API.

— Names the toolkit but no usage example given v3.1.0

└──▷ BREAKING ON UPGRADE

! The `agno_fs` filesystem table is now keyed by (namespace, user_id, path). A table created by an earlier release raises `SchemaOutdatedError` until migrated manually (migration never runs automatically). Run `python libs/agno/migrations/migrate_filesystem_postgres.py` (PostgreSQL) or `python libs/agno/migrations/migrate_filesystem_sqlite.py` (SQLite) with the application stopped.

! MCPConfig(tools=[...]) no longer serves built-in `default_tools` or lifecycle tools (`continue_run` / `cancel_run`) automatically — both now default to False and must be opted in explicitly.

v3.1.0 NOTES ↗ LATEST STABLE

Agno v3.1 adds RBAC, a database-backed filesystem, AIMLAPITools, and makes MCP default tools opt-in.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v3.1.0 https://github.com/agno-agi/agno.git
# already have the repo? check out this version:
$ git checkout v3.1.0
```

└──▷ USE IT

Opt in to MCP lifecycle tools explicitly now that `default_tools` and `lifecycle_tools` are False by default in MCPConfig .

```python
from agno.mcp import MCPConfig

mcp = MCPConfig(
    tools=["my_custom_tool"],
    default_tools=True,
    lifecycle_tools=True,
)
```

Adds `agno.os.authz` package with role-based access control (RBAC) for AgentOS — includes a role store, scope policy, audit log, user directory, admin router, and pluggable authorization engines (`native` and fine-grained `fga`).

Adds `agno.fs.DbFileSystem` backed by an `agno_fs` table with `/filesystem` routes for listing, reading, and managing files.

Adds `AIMLAPITools` toolkit for image, video, speech, and transcription via the AI/ML API.

Refreshes MCP server config and built-in MCP auth handling.

MCPConfig(tools=[...]) now publishes exactly the tools listed — `default_tools` and `lifecycle_tools` default to `False` , making both opt-in.

└──▷ BREAKING ON UPGRADE

! The `agno_fs` filesystem table is now keyed by (namespace, user_id, path). A table created by an earlier release raises `SchemaOutdatedError` until migrated manually (migration never runs automatically). Run `python libs/agno/migrations/migrate_filesystem_postgres.py` (PostgreSQL) or `python libs/agno/migrations/migrate_filesystem_sqlite.py` (SQLite) with the application stopped.

! MCPConfig(tools=[...]) no longer serves built-in `default_tools` or lifecycle tools (`continue_run` / `cancel_run`) automatically — both now default to `False` and must be opted in explicitly.

WAS THIS USEFUL? 👍 👎

Stacklok Mecatl

Feedback ⓘ v0.0.44 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 v0.0.44 adds read-only inspection of jsonlstore event logs, faster live tool-result streaming ahead of batch drain, and a new kind-testing skill for mecak8s.

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

01 **Read-only access mode for jsonlstore** NEW 50

Adds a read-only access mode to existing `jsonlstore` stores, enabling inspection of persisted event logs without taking a write lock.

— Names the store but no API or flag to invoke it v0.0.44

02 **mecak8s kind-testing skill** NEW 40

Adds a skill for `mecak8s` kind testing, enabling local Kubernetes cluster validation within the harness.

— Names the skill but no usage or command shown v0.0.44

03 **Live tool-result streaming before batch drain** IMPROVED 35

Publishes live tool-result availability before the canonical batch drain completes, reducing latency for streaming agent output.

— Describes benefit but no mechanism or interface detail v0.0.44

v0.0.44 NOTES ↗ LATEST STABLE

Mecatl v0.0.44 adds live tool-result streaming before batch drain, read-only jsonlstore access, and a mecak8s kind-testing skill.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.0.44 https://github.com/stacklok/mecatl.git
# already have the repo? check out this version:
$ git checkout v0.0.44
```

Adds read-only access mode to existing `jsonlstore` stores, enabling inspection of persisted event logs without a write lock.

> Publishes live tool-result availability before the canonical batch drain completes, reducing latency for streaming agent output.
>
> Adds a skill for `mecak8s` kind testing, enabling local Kubernetes cluster validation within the harness.

WAS THIS USEFUL?  

◆ MCP Servers & Tooling

gram

Feedback ⓘ server@2.15.0 NOTES ↗ RANK

Gram provides MCP servers and AI-agent tools from APIs with access controls and observability.

> Gram shipped MCP-scoped guardrails and risk-policy targeting for its built-in Platform MCP toolsets, deeper MCP-server filtering and visibility across Risk Events and Watchdog, Observability plugin credential rotation, and new issuer-listing API filters, alongside Access Hub, identity-provider, and dashboard usability improvements.

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

01 Device Agent Setup org credential copy buttons NEW

85

Adds `org_slug` and `org_token` copy buttons to the Device Agent Setup tab's Organization values card, so admins can retrieve both values without stepping through a platform walkthrough; once a token exists, the mint button reads Re-generate token and creates an additional token rather than rotating, so deployed device tokens stay valid until explicitly revoked under Settings → API Keys.

– exact field names and settings path given dashboard@0.129.0

02 MCP server filtering and visibility in Risk Events and Watchdog

IMPROVED

81

The `list_risk_findings` MCP tool now supports filtering findings by MCP server. Watchdog signals can be filtered by MCP server with per-server finding counts and reporting of which servers and tools each signal was observed on, including a Server group-by mode and a server/tool line on signals. Risk Events and Watchdog display findings from MCP tool calls with server, tool, outcome, and user context instead of an untitled session, and Risk Events rows now show the tool and enforcement outcome; the MCP server filter on Risk Events is always visible while the remaining additions are gated behind the MCP-scoped policies flag. Risk Events also now retains findings from disabled policies, marking them inactive instead of clearing the page.

– named tool and detailed behavior, but no exact navigation path dashboard@0.129.0 server@2.15.0

03 Issuer listing API query filters NEW

80

Adds `search` and `upstream_host` query filters to the issuer listing API, and a `tier` filter to the Remote Identity Providers issuer listing, e.g. `GET /issuers?search=myorg&upstream_host=ai.speakeasy.com&tier=platform`.

Filter the issuer listing to a specific upstream host and tier when auditing which identity providers are registered at the platform level.

```
$ GET /issuers?
search=myorg&upstream_host=ai.speakeasy.com&tier=platform
```

— named params with a runnable example request `server@2.15.0`

04 **Wildcard admission default change for workload issuers** BREAKING

80

`workload_issuers.allow_wildcard_admission` now defaults to on; existing issuers that had it off and relied on that gate to block wildcard rules will pass wildcard rules through unless the column is explicitly cleared (clearing it still makes every wildcard rule under that issuer inert immediately). Registering a workload issuer no longer presents a wildcard-admission permission switch in the UI; the field remains in the API.

— exact column name and migration behavior given `server@2.15.0`

05 **Platform MCP skill insight tools** NEW

73

Adds `list_skill_insights` and `compare_skill_versions` Platform MCP tools to rank and compare project skills by activations, sampled efficacy, session cost, and estimated time saved.

— named tools and metrics, no example call shown `server@2.15.0`

06 **Guardrails configuration in MCP server flow** NEW

70

Adds MCP-scoped guardrails configuration to the server create/edit flow: an opt-in Guardrails section when adding a remote or tunneled server, a skippable Guardrails step in the catalog install, and a Guardrails tab on the MCP server page listing scoped and inherited policies and recent risk events, gated behind the MCP-scoped policies flag.

— UI entry points named but no exact config key `dashboard@0.129.0`

07 **Observability plugin credential rotation** NEW

63

Organization admins can mint a replacement hooks-scoped ingest key for the Observability plugin, choosing either immediate revocation or a 7-day grace-window revocation of the prior key.

— mechanism and window named, no UI path or endpoint given `server@2.15.0`

THINNER COVERAGE BELOW

08 **Access Hub machine tags and platform descriptions** NEW

55

Allowed machines in the Access Hub can carry optional tags, stored on the admission with the same limits as trusted-platform tags; trusted platforms accept an optional description of up to 500 characters, shown on each platform card in place of the issuer URL.

— limit and location named, no exact field name `server@2.15.0`

09 **Identity provider picker search and on-demand loading** IMPROVED

55

The identity provider picker on remote MCP server settings now searches on the server and loads providers on demand, organized by project, organization, and platform tier with per-tier more rows.

— behavior described, no exact navigation path `server@2.15.0`

10 **Ranked chart type in Explore** NEW

55

Adds a Ranked chart type to Explore: full-window results rendered as horizontal bars (one per group, largest first, ranked by the ordered measure), matching the style used on MCP & Tools.

— UI feature described, no exact navigation steps `dashboard@0.129.0`

11 **Remote session client compatibility mode migration** NEW

45

Platform admins can now see, toggle, and migrate remote session clients running in compatibility mode with the legacy callback URL.

— admin action described, no exact UI path `server@2.15.0`

12 **MCP-scoped risk policy targeting** NEW

42

MCP-scoped risk policies can now target Gram's built-in Platform MCP toolsets and their individual tools.

— thin one-line description, no config or UI detail `server@2.15.0`

13 **Remote identity provider scope override editing** NEW

42

Supports viewing and editing a remote identity provider's scope override, with warnings shown where a client's scopes are edited and the override makes them inert.

— mechanism described but no UI path or field name `server@2.15.0`

14 **Connectivity check skip for direct remote MCP servers** NEW

40

Remote MCP servers that clients connect to directly can now be saved without passing the connectivity check, since those servers are often unreachable from Gram's

infrastructure.

— behavior described, no flag or setting named `dashboard@0.129.0`

└──▷ BREAKING ON UPGRADE

! `workload_issuers.allow_wildcard_admission` now defaults to on; existing issuers that had it off and relied on that gate to block wildcard rules will pass wildcard rules through unless the column is explicitly cleared (clearing it still makes every wildcard rule under that issuer inert immediately).

! Registering a workload issuer no longer presents a wildcard-admission permission switch in the UI; the field remains in the API.

`server@2.15.0` NOTES ↗ **LATEST** STABLE

Gram 2.15.0 adds Observability plugin credential rotation, MCP server filtering for Watchdog, and richer MCP risk-policy targeting.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch server@2.15.0 https://github.com/speakeasy-api/gram.git
# already have the repo? check out this version:
$ git checkout server@2.15.0
```

└──▷ TRY IT

Filter the issuer listing to a specific upstream host and tier when auditing which identity providers are registered at the platform level.

```
$ GET /issuers?search=myorg&upstream_host=ai.speakeasy.com&tier=platform
```

Adds `search` and `upstream_host` query filters to the issuer listing API, and a `tier` filter to the Remote Identity Providers issuer listing.

Adds `list_skill_insights` and `compare_skill_versions` Platform MCP tools to rank and compare project skills by activations, sampled efficacy, session cost, and estimated time saved.

Adds `list_risk_findings` MCP tool support for filtering findings by MCP server.

Enables Observability plugin credential rotation: organization admins can mint a replacement hooks-scoped ingest key with an immediate or 7-day grace-window revocation of the prior key.

MCP-scoped risk policies can now target Gram's built-in Platform MCP toolsets and their individual tools.

+7 more

└──▷ **BREAKING ON UPGRADE**

! `workload_issuers.allow_wildcard_admission` now defaults to on; existing issuers that had it off and relied on that gate to block wildcard rules will pass wildcard rules through unless the column is explicitly cleared (clearing it still makes every wildcard rule under that issuer inert immediately).

! Registering a workload issuer no longer presents a wildcard-admission permission switch in the UI; the field remains in the API.

`dashboard@0.129.0` NOTES ↗ STABLE

gram: Adds `org_slug` and `org_token` copy buttons to the Device Agent Setup tab's Organization values card, so admins can retrieve both values without stepping through a platform walkthrough; once a token exists, the mint button reads `Re-generate token` and creates an additional token rather than rotating, so deployed device tokens stay valid until explicitly revoked under Settings → API Keys.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch dashboard@0.129.0
https://github.com/speakeasy-api/gram.git
# already have the repo? check out this version:
$ git checkout dashboard@0.129.0
```

Adds a Ranked chart type to Explore: full-window results rendered as horizontal bars (one per group, largest first, ranked by the ordered measure), matching the style used on MCP & Tools.

Adds MCP-scoped guardrails configuration to the server create/edit flow: an opt-in Guardrails section when adding a remote or tunneled server, a skippable Guardrails step in the catalog install, and a Guardrails tab on the MCP server page listing scoped and inherited policies and recent risk events (gated behind the MCP-scoped policies flag).

Adds an MCP server filter with per-server finding counts to Risk Events and Watchdog, a Server group-by mode and server/tool line on Watchdog signals, and shows the tool and enforcement outcome on Risk Events rows (MCP server filter on Risk Events is always visible; remaining features gated behind the MCP-scoped policies flag).

Risk Events now retains findings from disabled policies, marking them inactive, so disabling a policy no longer clears the page; findings from MCP-scoped policies always appear.

Remote MCP servers that clients connect to directly can be saved without passing the connectivity check, since those servers are often unreachable from Gram's

infrastructure.

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Agentic Infrastructure

OpenRouter

Feedback ⓘ snapshot-20261001 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 confidence-threshold routing, letting requests stay on a cheap model by default and automatically escalate to a more capable one when confidence is low.

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

01 **Confidence-based model escalation routing** NEW 63

OpenRouter can keep most requests on a cheaper model and automatically escalate only low-confidence requests to a more capable one. Escalation is driven by a numeric confidence field returned via structured outputs, with thresholds tunable against error rates, accuracy, cost, and latency measured from the caller's own traffic.

— Explains mechanism and tuning inputs but no exact API field name or endpoint given

snapshot-20261001

> snapshot-20261001 NOTES ↗ LATEST STABLE
>
> OpenRouter adds confidence-threshold routing to escalate uncertain requests from cheaper to more capable models.
>
> Adds confidence-based model escalation routing that keeps most requests on a cheap model and automatically escalates only low-confidence requests to a more capable one.
>
> Supports a numeric confidence field via structured outputs to drive escalation decisions, with thresholds tunable against error rates, accuracy, cost, and latency from your own traffic.

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.8.105 NOTES ↗ RANK

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

> HyperFrames shipped a new Graphite light theme for Studio alongside a set of host app extensibility APIs letting embedding apps customize the Studio dock, canvas z-order, and clip context menus.

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

01 **Host app extensibility APIs for Studio UI** NEW 68

Host apps can now customize Studio's UI surface through three new APIs: the `useDomEditZOrder` hook lets a host app apply the Studio canvas menu's z-order to layers programmatically; the Studio dock can be mounted by a host app with its own panels, storage key, and width; and Studio hosts can add their own items to a clip's right-click context menu.

– Names the hook and dock options but no code samples or full API signatures v0.8.102 v0.8.101 v0.8.98

THINNER COVERAGE BELOW

02 **Graphite colour system with light theme in Studio** NEW 35

Studio gains a Graphite colour system with a light theme that passes contrast requirements.

– Names the theme but gives no usage path or technical detail v0.8.105

v0.8.105 NOTES ↗ LATEST STABLE

HyperFrames v0.8.105 adds a Graphite colour system with a light theme to Studio.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.8.105 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.105
```

v0.8.102 NOTES ↗ STABLE

HyperFrames v0.8.102 lets host apps apply canvas z-order programmatically via `useDomEditZOrder`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.8.102 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.102
```

v0.8.101 NOTES ↗ STABLE

HyperFrames v0.8.101 lets host apps mount the Studio dock with custom panels, storage keys, and widths.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.8.101 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.101
```

v0.8.98 NOTES ↗ STABLE

HyperFrames v0.8.98 lets Studio hosts add custom items to the clip right-click menu.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.8.98 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.98
```

WAS THIS USEFUL? 👍 👎

Anthropic Python SDK

Feedback ⓘ v1.11.0 NOTES ↗ RANK

The Anthropic Python SDK provides typed Python clients for Claude APIs, streaming responses, and tool use.

> Across v1.10.0 and v1.11.0 the Anthropic Python SDK expanded the Admin API with enterprise spend limits, analytics, RBAC and Plugins support, gave Managed Agents sessions refusal detection and richer error types, and added transport/token detail to MCP Tunnel while promoting Organization API endpoints to GA.

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

01 Refusal stop reason in Managed Agents sessions NEW

85

Adds a `refusal` stop reason and `stop_details` to Managed Agents session idle events, surfacing when a session ends due to a model refusal.

Detect a model refusal during a Managed Agents session stream so you can log or escalate policy violations.

python

```
with anthropic.beta.sessions.events.stream(session.id) as
stream:
    for event in stream:
        if event.type == 'session.status_idle':
            if hasattr(event, 'stop_details') and
event.stop_details.stop_reason == 'refusal':
                print('Session ended with refusal:',
event.stop_details)
            break
```

— Names the fields and shows a runnable detection snippet v1.10.0

02 MCP Tunnel transport object and relay token NEW

60

Adds a read-only `transport` object to MCP Tunnel and returns a one-time relay `token` in the tunnel create response (MCP tunnels beta).

— Names exact response fields though no full endpoint path v1.10.0

THINNER COVERAGE BELOW

03 Admin API enterprise controls: spend limits, analytics, RBAC

NEW

40

Adds a list spend limits endpoint to the API, and extends the Admin API with Claude Enterprise analytics, spend limits, and RBAC groups and roles, including per-user usage and cost reports in Admin API analytics.

— Names capabilities added but no endpoint paths or request shapes `v1.10.0` `v1.11.0`

04 **Plugins and Plugin Marketplaces in Admin API** NEW 40

Adds Plugins and Plugin Marketplaces to the Admin API, plus the ability to remove a plugin's org-wide installation setting via the Admin API.

— Describes the additions but no endpoint or flag given `v1.10.0`

05 **Organization API endpoints promoted to GA** IMPROVED 30

Promotes Organization API endpoints from beta to GA.

— States the status change but no endpoint list or migration notes `v1.10.0`

06 **Repository error type in Managed Agents sessions** NEW 25

Adds `repository` error types to Managed Agents session errors.

— Bare naming of a new error type, no further context `v1.10.0`

v1.11.0 NOTES ↗ LATEST STABLE

Anthropic Python SDK v1.11.0 adds a list spend limits endpoint.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.11.0
https://github.com/anthropics/anthropic-sdk-python.git
# already have the repo? check out this version:
$ git checkout v1.11.0
```

v1.10.0 NOTES ↗ STABLE

Anthropic Python SDK v1.10.0 adds Claude Enterprise analytics, RBAC, spend limits, Plugins, and MCP tunnel transport to the Admin API.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.10.0
https://github.com/anthropics/anthropic-sdk-python.git
# already have the repo? check out this version:
$ git checkout v1.10.0
```

└──▷ USE IT

Detect a model refusal during a Managed Agents session stream so you can log or escalate policy violations.

```python
with anthropic.beta.sessions.events.stream(session.id) as stream:
    for event in stream:
        if event.type == 'session.status_idle':
            if hasattr(event, 'stop_details') and
event.stop_details.stop_reason == 'refusal':
                print('Session ended with refusal:',
event.stop_details)
            break
```

Adds `refusal` stop reason and `stop_details` to Managed Agents session idle events, surfacing when a session ends due to a model refusal.

Adds Claude Enterprise analytics, spend limits, and RBAC groups and roles to the Admin API.

Adds per-user usage and cost reports to the Admin API analytics.

Adds Plugins and Plugin Marketplaces to the Admin API.

Adds `repository` error types to Managed Agents session errors.

+3 more

WAS THIS USEFUL?  

NVIDIA Triton Inference Server

Feedback ⓘ v2.73.0

NOTES ↗ RANK

Triton Inference Server is an open-source platform for deploying and serving AI models at scale across CPUs and GPUs.

> Triton v2.73.0 adds initial support for NVIDIA's next-generation Rubin GPU architecture and restores enqueue-time slot reservation for `preserve_ordering`.

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

01 **Initial NVIDIA Rubin GPU architecture support** NEW 40

Triton adds initial support for the NVIDIA Rubin GPU architecture, delivered via the 26.09 DLFW (Deep Learning Framework) containers.

— Names the GPU architecture and container version but no usage detail v2.73.0

02 **Restored preserve_ordering slot reservation** IMPROVED 35

Triton restores `preserve_ordering` enqueue-time slot reservation, a behavior that reserves ordering slots at the time a request is enqueued.

— Names the flag but gives no mechanism or fix context v2.73.0

v2.73.0 NOTES ↗ LATEST STABLE

Triton v2.73.0 adds initial NVIDIA Rubin GPU architecture support and restores `preserve_ordering` enqueue-time slot reservation.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.73.0 https://github.com/triton-inference-server/server.git
# already have the repo? check out this version:
$ git checkout v2.73.0
```

WAS THIS USEFUL? 👍 👎

Google Gen AI JavaScript SDK

Feedback ⓘ v2.25.0

NOTES ↗ RANK

The Google Gen AI JavaScript SDK provides TypeScript and JavaScript interfaces to Gemini APIs and live model sessions.

> v2.25.0 of the Google Gen AI JavaScript SDK adds environment-variable client construction, GCS metrics export for tuning jobs, labels on GenerateContent and live sessions, and two new TTS models.

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

01 **Labels on GenerateContent and LiveClientSetup** NEW 75

Adds `labels` support to `GenerateContent` requests for the Gemini API, and adds a `labels` field to `LiveClientSetup` for live session labeling, letting requests be tagged (e.g. by team or environment) for usage tracking.

Tag GenerateContent requests with cost-center or team labels so you can slice usage reports by project or team in the Gemini API console.

typescript

```typescript
import {GoogleGenAI} from '@google/genai';

const ai = new GoogleGenAI({apiKey:
process.env.GOOGLE_API_KEY});

const response = await ai.models.generateContent({
  model: 'gemini-flash-latest',
  contents: 'Analyze this log for anomalies.',
  labels: {
    team: 'security-ops',
    environment: 'production',
  },
});
console.log(response.text);
```

— Includes runnable code example tagging requests with labels v2.25.0

02 **Environment-variable client initialization** NEW 65

A dedicated environments module with `from_environment` support lets you construct a GoogleGenAI() client directly from environment variables `GOOGLE_API_KEY`, `GOOGLE_GENAI_USE_ENTERPRISE`, `GOOGLE_CLOUD_PROJECT`, and `GOOGLE_CLOUD_LOCATION`.

— Names exact env vars but no full code sample v2.25.0

THINNER COVERAGE BELOW

03 **GCS metrics export for tuning jobs** NEW 45

Adds a `gcs_metrics_uri` field to export tuning job metrics to Google Cloud Storage.

— Names the field but not its usage or format v2.25.0

04 **Gemini 3.8 Flash TTS models** NEW 40

Adds Gemini 3.8 Flash TTS and Gemini 3.8 Flash Lite TTS models to the SDK.

— Names the models but no usage detail v2.25.0

v2.25.0 NOTES ↗ LATEST STABLE

v2.25.0 adds environment-variable client init, GCS metrics export for tuning jobs, labels for GenerateContent and LiveClientSetup, and two new TTS models.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.25.0 https://github.com/googleapis/js-genai.git
# already have the repo? check out this version:
$ git checkout v2.25.0
```

└──▷ USE IT

Tag GenerateContent requests with cost-center or team labels so you can slice usage reports by project or team in the Gemini API console.

```typescript
import {GoogleGenAI} from '@google/genai';

const ai = new GoogleGenAI({apiKey: process.env.GOOGLE_API_KEY});

const response = await ai.models.generateContent({
  model: 'gemini-flash-latest',
  contents: 'Analyze this log for anomalies.',
  labels: {
    team: 'security-ops',
    environment: 'production',
  },
});
console.log(response.text);
```

Adds `gcs_metrics_uri` field to export tuning job metrics to Google Cloud Storage.

Adds `labels` support to `GenerateContent` requests for the Gemini API.

Adds `labels` field to `LiveClientSetup` for live session labeling.

Adds a dedicated environments module and `from_environment` support, enabling GoogleGenAI() client construction from environment variables `GOOGLE_API_KEY`, `GOOGLE_GENAI_USE_ENTERPRISE`, `GOOGLE_CLOUD_PROJECT`, and `GOOGLE_CLOUD_LOCATION`.

Adds Gemini 3.8 Flash TTS and Gemini 3.8 Flash Lite TTS models to the SDK.

WAS THIS USEFUL? 👍 👎

Google Gen AI Python SDK

Feedback ⓘ v2.26.0 NOTES ↗

RANK

The Google Gen AI Python SDK provides Python interfaces to Gemini model generation, streaming, tools, and embeddings.

> v2.26.0 adds a dedicated environments module for client construction, a `gcs_metrics_uri` export option for tuning jobs, `labels` support across GenerateContent and live sessions, and two new Gemini 3.8 Flash TTS models.

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

01 **GCS export for tuning job metrics** NEW 75

Adds a `gcs_metrics_uri` field to tuning job configuration (`CreateTuningJobConfig`) for exporting fine-tuning job metrics to a Google Cloud Storage bucket, enabling analysis in tools like BigQuery or custom dashboards.

Export fine-tuning job metrics to a GCS bucket for analysis in BigQuery or custom dashboards.

python

```python
tuning_job = client.tunings.tune(
    base_model='gemini-flash-latest',
    training_dataset=dataset,
    config=types.CreateTuningJobConfig(
        gcs_metrics_uri='gs://my-bucket/tuning-metrics/'
    )
)
```

— Names exact config field and shows runnable tuning example v2.26.0

02 **Labels field on GenerateContent and LiveClientSetup** NEW 60

Adds a `labels` field to both `GenerateContent` requests and `LiveClientSetup` for live session configuration, allowing requests to be tagged (e.g. with cost-center or environment labels) for billing attribution and audit filtering.

Tag GenerateContent requests with cost-center or environment labels for billing attribution and audit filtering.

```python
response = client.models.generate_content(
    model='gemini-flash-latest',
    contents='Summarize this document.',
    config=types.GenerateContentConfig(
        labels={'team': 'security-ops', 'env': 'prod'}
    )
)
```

— Two named surfaces grouped with a runnable code example v2.26.0

THINNER COVERAGE BELOW

03 **Environments module and from_environment client init** NEW

42

Adds a dedicated `environments` module and `from_environment` support for constructing clients from environment configuration.

— Names the module and function but no usage shown v2.26.0

04 **Gemini 3.8 Flash TTS models** NEW

30

Adds Gemini 3.8 Flash TTS and Gemini 3.8 Flash Lite TTS models to the SDK.

— Names two models but no usage or mechanism given v2.26.0

v2.26.0 NOTES ↗ LATEST STABLE

v2.26.0 adds a dedicated environments module, `from_environment` client init, `gcs_metrics_uri` for tuning job export, and labels support for GenerateContent and LiveClientSe

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.26.0 https://github.com/googleapis/python-genai.git
# already have the repo? check out this version:
$ git checkout v2.26.0
```

└──▷ USE IT

Export fine-tuning job metrics to a GCS bucket for analysis in BigQuery or custom dashboards.

```python
tuning_job = client.tunings.tune(
    base_model='gemini-flash-latest',
    training_dataset=dataset,
    config=types.CreateTuningJobConfig(
        gcs_metrics_uri='gs://my-bucket/tuning-metrics/'
    )
)
```

Tag GenerateContent requests with cost-center or environment labels for billing attribution and audit filtering.

```python
response = client.models.generate_content(
    model='gemini-flash-latest',
    contents='Summarize this document.',
    config=types.GenerateContentConfig(
        labels={'team': 'security-ops', 'env': 'prod'}
    )
)
```

Adds `gcs_metrics_uri` field to tuning job configuration for exporting tuning job metrics to Google Cloud Storage.

Adds a dedicated `environments` module and `from_environment` support for constructing clients from environment configuration.

Supports `labels` field on `GenerateContent` requests for the Gemini API.

Supports `labels` field on `LiveClientSetup` for live session configuration.

Adds Gemini 3.8 Flash TTS and Gemini 3.8 Flash Lite TTS models to the SDK.

WAS THIS USEFUL? 👍 👎

Eigen Labs Darkbloom

Feedback ⓘ v0.9.14 NOTES ↗ CODE ↗

RANK

Private Inference Network on Idle Macs

> Darkbloom v0.9.14 adds an environment variable to control MiMo SSD prefix caching on provider nodes.

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

01 MiMo prefix caching override via environment variable NEW

86

Darkbloom forwards the `DARKBLOOM_MIMO_COMPLETE_PREFIX` environment variable to the launchd provider job, letting operators override MiMo COMPLETE-prefix SSD caching per-deployment: setting it to exact `1` enables caching, any other non-empty value disables it, and leaving it unset falls back to the model default.

Roll back MiMo prefix caching on a provider node that is hitting SSD pressure — set the override before restarting so the new plist takes effect.

```
$ DARKBLOOM_MIMO_COMPLETE_PREFIX=0 darkbloom start
```

— Names exact env var, enable/disable values and effect; includes runnable command.

v0.9.14

v0.9.14 NOTES ↗ CODE ↗ LATEST STABLE

Darkbloom v0.9.14 enables MiMo SSD prefix caching by default, controlled via `DARKBLOOM_MIMO_COMPLETE_PREFIX`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.9.14 https://github.com/Layr-Labs/d-inference.git
# already have the repo? check out this version:
$ git checkout v0.9.14
```

└──▷ TRY IT

Roll back MiMo prefix caching on a provider node that is hitting SSD pressure — set the override before restarting so the new plist takes effect.

```
$ DARKBLOOM_MIMO_COMPLETE_PREFIX=0 darkbloom start
```

WAS THIS USEFUL? 👍 👎

Anthropic TypeScript SDK

Feedback ⓘ sdk-v0.130.0

NOTES ↗ RANK

The Anthropic TypeScript SDK provides typed JavaScript and TypeScript clients for Claude APIs, streaming, and tool use.

> The Anthropic TypeScript SDK's Admin API gains Claude Enterprise analytics, spend limits and RBAC, a Plugin Marketplace, and improved MCP tunnel transport, alongside refinements to Managed Agents session events.

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

01 **MCP Tunnel transport and relay token** IMPROVED 55

Adds a read-only `transport` object to MCP Tunnel responses and returns a one-time relay `token` in the tunnel create response (MCP tunnels beta).

— Names exact response fields `transport` and `token` sdk-v0.130.0

02 **Enterprise analytics, spend limits and RBAC in Admin API** NEW 50

The Admin API adds Claude Enterprise analytics, spend limits, and RBAC groups and roles, plus per-user usage and cost reports in Admin API analytics.

— Names features but no endpoint paths or field names sdk-v0.130.0

03 **Managed Agents session event and error details** NEW 45

Adds `stop_details` and a `refusal` stop reason to Managed Agents session idle events, and adds repository error types to Managed Agents session errors.

— Names fields but no code example or endpoint sdk-v0.130.0

04 **Plugins and Plugin Marketplaces in Admin API** NEW 40

Adds Plugins and Plugin Marketplaces to the Admin API, along with the ability to remove a plugin's org-wide installation setting.

— Describes additions without concrete endpoint or config names sdk-v0.130.0

05 **Organization API endpoints promoted to GA** IMPROVED 25

Promotes Organization API endpoints from beta to GA.

— No endpoint names or migration detail given sdk-v0.130.0

sdk-v0.130.0 NOTES ↗ LATEST STABLE

Anthropic TypeScript SDK adds Claude Enterprise analytics, spend limits, RBAC, Plugin Marketplace, and MCP tunnel transport to the Admin API.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch sdk-v0.130.0
https://github.com/anthropics/anthropic-sdk-typescript.git
# already have the repo? check out this version:
$ git checkout sdk-v0.130.0
```

Adds `stop_details` and a `refusal` stop reason to Managed Agents session idle events.

Adds repository error types to Managed Agents session errors.

Adds a read-only `transport` object to MCP Tunnel responses and returns a one-time relay `token` in the tunnel create response (MCP tunnels beta).

Adds Claude Enterprise analytics, spend limits, and RBAC groups and roles to the Admin API.

Adds per-user usage and cost reports to the Admin API analytics.

+3 more

WAS THIS USEFUL? 👍 👎

Hugging Face Transformers

Feedback ⓘ v5.18.0 NOTES ↗

RANK

Transformers: the model-definition framework for state-of-the-art machine learning models in text, vision, audio, and multimodal models, for both inference and training.

> Transformers v5.18.0 ships four new model architectures — the Nemotron 3 streaming diarization model, the NemotronH Omni and HyperCLOVAX Vision V2 multimodal reasoning models, and the GTE text-embedding encoder — alongside MoE support in the GGUF loader, FSDP2 combined with expert parallelism, MUSA support for Flash Attention 2, a config option to untie hidden states, and six breaking changes spanning ROCm, DETR, indexers, vLLM, DINO, and the Kernels package.

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

01 GTE bidirectional text embedding model NEW

85

`GTE` is a BERT-style bidirectional encoder using RoPE in place of absolute position embeddings, a gated MLP, and post-residual layer normalization, backing Alibaba `gte-*-v1.5`, `gte-multilingual-*`, `gte-en-mlm-*`, and Snowflake `snowflake-arctic-embed-m-v2.0` checkpoints. It can be loaded via `AutoModel.from_pretrained` to generate dense text embeddings for retrieval or semantic search, as shown with `Alibaba-NLP/gte-multilingual-base`.

Load a GTE embedding model to generate dense text representations for retrieval or semantic search.

python

```python
from transformers import AutoModel, AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained('Alibaba-NLP/gte-multilingual-base')
model = AutoModel.from_pretrained('Alibaba-NLP/gte-multilingual-base')

inputs = tokenizer(['Hello world', 'Bonjour monde'], return_tensors='pt', padding=True)
outputs = model(**inputs)
embeddings = outputs.last_hidden_state[:, 0]  # CLS token
```

— Full architecture, checkpoint names, and runnable embedding example. v5.18.0

02 Nemotron 3 streaming speaker diarization model NEW

65

`Nemotron3Diarization` is an open-weight streaming speaker diarization model supporting up to 8 speakers, with both streaming and offline inference modes, configurable latency profiles from 80 ms to 30.4 s, and configurable output frame resolution in multiples of 10 ms. Chunked inference allows unlimited audio duration.

— Numbers for speakers, latency, frame resolution but no usage example. v5.18.0

---

03 **NemotronH Omni multimodal reasoning model** NEW 65

`NemotronH Omni` is a multimodal reasoning model from NVIDIA that combines the NemotronH hybrid Mamba-Transformer LLM with a RADIO vision encoder and an optional Parakeet-based sound encoder, enabling joint reasoning over text, images, video, and audio via `<image>`, `<video>`, and `<audio>` context tokens.

— Names architecture components and tokens but no code example. v5.18.0

THINNER COVERAGE BELOW

04 **HyperCLOVAX Vision V2 multimodal model** NEW 55

`HyperCLOVAX Vision V2` is a multimodal vision-language model from NAVER that combines the HyperClovaX LLM with a Qwen2.5-VL vision encoder, supporting text, image, and video inputs with chain-of-thought reasoning via built-in `<think>...</think>` tokens.

— Names architecture and tokens but lacks usage example. v5.18.0

---

05 **FSDP2 with expert parallelism** NEW 45

FSDP2 can now be combined with expert parallelism via a 2-D (fsdp, tp) device mesh.

— Names device mesh shape but no concrete usage shown. v5.18.0

---

06 **Untie hidden_states from last_hidden_state** NEW 45

A new model config option allows untying `hidden_states[-1]` from `last_hidden_state`.

— Names config field but no example or default given. v5.18.0

---

07 **MoE support in GGUF integration** NEW 40

The GGUF integration now supports mixture-of-experts models, enabling GGUF-format loading for MoE architectures including Qwen3.8.

— Names target model but no mechanism or command shown. v5.18.0

---

08 **ROCm FA3 routing change for gpt-oss** BREAKING 40

On ROCm, FA3 is now routed to aiter-flash-attn for gpt-oss, and ROCm fixtures have been regenerated; existing ROCm setups using FA3 with gpt-oss may see changed behavior.

– Names routing change but no migration steps given. v5.18.0

09 **vLLM video token counting change** BREAKING 40

vLLM video token counting for Transformers-backend video inputs has changed (Part 1); vLLM integrations passing video inputs may produce incorrect token counts until updated.

– Notes behavior change but no specifics or fix given. v5.18.0

10 **DETR image processing speedup and interface change** BREAKING 35

DETR image processing has been sped up, with interface changes that may require review of existing DETR image processing pipelines.

– Notes interface change but no specifics on what changed. v5.18.0

11 **Kernels package version bump** BREAKING 30

The Kernels package version has been bumped; downstream code pinning the previous Kernels version may be incompatible.

– Version bump noted but no version numbers given. v5.18.0

12 **MUSA support for Flash Attention 2** NEW 25

Flash Attention 2 now supports the MUSA backend.

– One-line hardware support note, no detail. v5.18.0

13 **Indexer layer_type value remapping** BREAKING 25

Indexer `layer_type` values have been remapped, breaking configurations that rely on the previous `layer_type` indexer values.

– Notes remapping but no old/new values given. v5.18.0

14 **DINO model modernization** BREAKING 20

DINO models have been brought up to modern standards, which may require migration of existing DINO model checkpoints or code.

– Vague modernization note, no concrete changes listed. v5.18.0

└──▷ ALSO FROM THESE RELEASES

.

└──▷ BREAKING ON UPGRADE

- ! ROCm: FA3 is now routed to aiter-flash-attn for gpt-oss, and ROCm fixtures are regenerated — existing ROCm setups using FA3 with gpt-oss may see changed behavior.
- ! DETR image processing has been sped up with interface changes — existing DETR image processing pipelines may need review.
- ! Indexers `layer_type` values have been remapped — configurations relying on the previous `layer_type` indexer values will break.
- ! vLLM video token counting for Transformers backend video inputs has changed (Part 1) — vLLM integrations passing video inputs may produce incorrect token counts until updated.
- ! DINO models have been brought to modern standards — existing DINO model checkpoints or code may require migration.
- ! The `Kernels` package version has been bumped — downstream code pinning the previous `Kernels` version may be incompatible.

v5.18.0 NOTES ↗ LATEST STABLE

Adds Nemotron 3 Diarization, NemotronH Omni, HyperCLOVAX Vision V2, and GTE model support to Transformers v5.18.0.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v5.18.0
https://github.com/huggingface/transformers.git
# already have the repo? check out this version:
$ git checkout v5.18.0
```

└──▷ USE IT

Load a GTE embedding model to generate dense text representations for retrieval or semantic search.

```python
from transformers import AutoModel, AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained('Alibaba-NLP/gte-
multilingual-base')
model = AutoModel.from_pretrained('Alibaba-NLP/gte-multilingual-
base')

inputs = tokenizer(['Hello world', 'Bonjour monde'],
return_tensors='pt', padding=True)
outputs = model(**inputs)
embeddings = outputs.last_hidden_state[:, 0]  # CLS token
```

Adds `Nemotron3Diarization` — an open-weight streaming speaker diarization model supporting up to 8 speakers, both streaming and offline inference, configurable latency

profiles from 80 ms to 30.4 s, and configurable output frame resolution in multiples of 10 ms, with unlimited audio duration via chunked inference.

Adds `NemotronH Omni` — a multimodal reasoning model from NVIDIA that combines the NemotronH hybrid Mamba-Transformer LLM with a RADIO vision encoder and optional Parakeet-based sound encoder, supporting joint reasoning over text, images, video, and audio via `<image>`, `<video>`, and `<audio>` context tokens.

Adds `HyperCLOVAX Vision V2` — a multimodal vision-language model from NAVER combining the HyperClovaX LLM with a Qwen2.5-VL vision encoder, supporting text, image, and video inputs with chain-of-thought reasoning via built-in `<think>...</think>` tokens.

Adds `GTE` — a BERT-style bidirectional encoder using RoPE in place of absolute position embeddings, a gated MLP, and post-residual layer normalization, backing Alibaba `gte-*-v1.5`, `gte-multilingual-*`, `gte-en-mlm-*`, and Snowflake `snowflake-arctic-embed-m-v2.0` checkpoints.

Adds MoE support to the GGUF integration, enabling GGUF-format loading for mixture-of-experts models including Qwen3.8.

+3 more

└─▷ BREAKING ON UPGRADE

- ! ROCm: FA3 is now routed to aiter-flash-attn for gpt-oss, and ROCm fixtures are regenerated — existing ROCm setups using FA3 with gpt-oss may see changed behavior.
- ! DETR image processing has been sped up with interface changes — existing DETR image processing pipelines may need review.
- ! Indexers `layer_type` values have been remapped — configurations relying on the previous `layer_type` indexer values will break.
- ! vLLM video token counting for Transformers backend video inputs has changed (Part 1) — vLLM integrations passing video inputs may produce incorrect token counts until updated.
- ! DINO models have been brought to modern standards — existing DINO model checkpoints or code may require migration.
- ! The `Kernels` package version has been bumped — downstream code pinning the previous `Kernels` version may be incompatible.

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

MLX Serve

Feedback ⓘ v26.10.1 NOTES ↗ RANK

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

> mlx-serve v26.10.1 brings GGUF models onto the native MLX engine, adds a `/v1/decisions` endpoint, model aliases and several speculative-decoding tweaks, and delivers up to 66% faster decode and 51% faster prefill across supported models.

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

01 **Native GGUF execution on the MLX engine** NEW 90

The `--mlx-gguf` flag (or the Settings > Engines toggle) runs GGUF models — Qwen 3.5, 3.6, 3.8 dense/MoE and Gemma 4 — directly on the native MLX engine instead of falling back to llama.cpp, giving them prefix cache, batching, speculative decoding and tool-calling. The same release adds GGUF model loading support for Nemotron-H 3.5 MoE and Qwen3.8 Flash Next.

— Names the flag, engine mechanism, and specific models gaining support. v26.10.1

02 **`--ple-gpu` long-context performance flag** NEW 85

The `--ple-gpu` flag (or app toggle) boosts prefill 2-4% and decode +15% at 128k context; it is off by default because it requires a full-table memory allocation.

— Exact flag with measured before/after numbers and default state. v26.10.1

03 **`--mtp-greedy-tail` sampling flag** NEW 75

The `--mtp-greedy-tail` flag (or `"mtp_greedy_tail": true` set per-model in model settings) lets `--mtp-typical` accept longer runs on sampled requests.

— Exact flag and config key with a clear behavioural effect. v26.10.1

04 **Model aliases in model-settings.json** NEW 70

Adds an `"alias"` field in `model-settings.json` (or the Model Settings UI) for short model nicknames, usable in Ollama, for load/unload, and listed in `/v1/models`.

— Names the config field and every surface it appears on. v26.10.1

05 **`/v1/decisions` endpoint** NEW 65

Adds the `/v1/decisions` endpoint for fast yes/no and multiple-choice questions via the Kev model, alongside existing Laya support.

— Named endpoint but mechanism of the underlying models is thin. v26.10.1

06 **`mlx-serve` skill and `launch` for agent wiring** NEW

65

Adds an `mlx-serve` skill for agents launched from the app, or `mlx-serve launch` for wiring code to local MLX-Serve APIs.

– Names the skill and the runnable `mlx-serve launch` subcommand. v26.10.1

07 **App and console UI improvements** NEW

60

Adds sidebar folders for chats, agent threads and terminals; terminals now reopen at their last position; the memory meter splits weights, KV cache and other usage; and adds a Simplified Chinese translation and a light theme for the console and app alerts.

– Enumerates every UI change but each is a brief description. v26.10.1

08 **Model registration and Ollama compatibility reporting** IMPROVED

60

Models pulled via `mlx-serve pull` register under their `org/name` id, and Ollama's `/api/show` now reports thinking and vision flags for models not yet loaded.

– Names the command and the compatibility endpoint it affects. v26.10.1

THINNER COVERAGE BELOW

09 **EXL3 expert pack loading and default speculative drafting** NEW

55

Loads Sushi-format EXL3 expert packs for Qwen3.8 Flash Next directly with no conversion step, and enables speculative drafting by default for every model that supports it.

– Names the format and default behaviour but no flag given. v26.10.1

10 **Int8 prefill for 2-bit Bonsai 2** NEW

55

Adds an Int8 prefill setting for 2-bit Bonsai 2 to read prompts faster; it is off by default and carries a small accuracy cost.

– Named setting with tradeoff, but no exact flag or key given. v26.10.1

11 **Faster decode and prefill across models** IMPROVED

50

Delivers decode throughput up to 66% faster and prefill up to 51% faster across Gemma 4, Qwen3.8, Llama, Mistral and other models, with identical output to prior versions.

– Quantified gains across named model families but nothing to invoke. v26.10.1

12 **My Models — Check for Updates** NEW

50

Adds a My Models — Check for Updates feature that verifies models against Hugging Face before replacing any files.

— Named UI path and the safety check it performs. v26.10.1

13 **Qwen-Image-2.1 instruction-based editing** NEW 45

Enables Qwen-Image-2.1 to edit pictures from text instructions with up to 10 reference images.

— Describes the capability and a concrete limit, no invocation path. v26.10.1

14 **Drafter slot and bundled drafters for every model** NEW 45

Adds a drafter slot to every model in the app, with bundled drafters loading automatically, and 27B 6-bit/8-bit models gaining full-speed drafter support.

— Explains scope of the change but no configuration surface named. v26.10.1

15 **Reproducible output changes with DFlash** BREAKING 45

Seeded (reproducible) output with a DFlash drafter on Nemotron-H and dense Qwen3.5/3.8 (4, 6, and 8-bit) will differ from v26.9.6, because DFlash now matches normal decoding exactly.

— Names affected models and prior version for migration awareness. v26.10.1

16 **MLX-Serve Skills for Pi** NEW 40

Adds MLX-Serve Skills so Pi can build apps using local MLX-Serve APIs for audio, images and web app generation.

— States the capability but no concrete API names or steps. v26.10.1

17 **Qwen3.8-27B DFlash2 tree-of-drafts** NEW 40

Adds Qwen3.8-27B support for DFlash2 tree-of-drafts per round instead of single-path drafting.

— Names the mechanism but gives no way to enable or measure it. v26.10.1

18 **No stream freeze during long-prompt reads** IMPROVED 40

Eliminates chat stream freezes while another request reads a long prompt, keeping active streams moving during 32k-token prefill operations.

— Describes the fix and a concrete token count, no user action. v26.10.1

└──▷ BREAKING ON UPGRADE

! Seeded (reproducible) output with a DFlash drafter on Nemotron-H, dense Qwen3.5/3.8 (4, 6, and 8-bit) will differ from v26.9.6 because DFlash now matches normal decoding exactly.

v26.10.1 NOTES ↗ LATEST STABLE

mlx-serve v26.10.1 adds native GGUF-on-MLX, `--mlx-gguf`, `--ple-gpu`, `/v1/decisions`, model aliases, and up to 66% faster decode across Apple Silicon.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v26.10.1 https://github.com/ddalcu/mlx-serve.git
# already have the repo? check out this version:
$ git checkout v26.10.1
```

Adds `--mlx-gguf` flag (or Settings > Engines toggle) to run GGUF models — Qwen 3.5, 3.6, 3.8 dense/MoE and Gemma 4 GGUFs — directly on the native MLX engine instead of the llama.cpp fallback, giving them prefix cache, batching, speculative decoding and tool-calling.

Adds `--ple-gpu` flag (or app toggle) to boost prefill 2-4% and decode +15% at 128k context; off by default due to full-table memory requirement.

Adds `/v1/decisions` endpoint for fast yes/no and multiple-choice questions via the Kev model alongside existing Laya support.

Adds `"alias"` field in `model-settings.json` (or Model Settings UI) for short model nicknames usable in Ollama, load/unload, and listed in `/v1/models`.

Adds `--mtp-greedy-tail` flag (or `"mtp_greedy_tail": true` per-model in model settings) to let `--mtp-typical` accept longer runs on sampled requests.

+14 more

└──▷ BREAKING ON UPGRADE

! Seeded (reproducible) output with a DFlash drafter on Nemotron-H, dense Qwen3.5/3.8 (4, 6, and 8-bit) will differ from v26.9.6 because DFlash now matches normal decoding exactly.

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

Weaviate

Feedback ⓘ v1.38.18 NOTES ↗ RANK

Weaviate is an open-source vector database that stores both objects and vectors, allowing for the combination of vector search with structured filtering with the fault tolerance and scalability of a cloud-native database.

> Weaviate v1.38.18 adds metrics that track the requested consistency level across all API requests.

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

01 **Consistency-level tracking metrics** NEW 40

New metrics now track the requested consistency level across all API requests (feat(metrics)), giving operators visibility into which consistency level clients are requesting per call.

– Names the metric feature but no metric names or how to query them. v1.38.18

v1.38.18 NOTES ↗ LATEST STABLE

Weaviate v1.38.18 adds consistency-level tracking metrics across all API requests.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.38.18
https://github.com/weaviate/weaviate.git
# already have the repo? check out this version:
$ git checkout v1.38.18
```

Tracks requested consistency level in all API requests via new metrics (feat(metrics)).

WAS THIS USEFUL? 👍 👎

okf-memory OKF Agent Memory

Feedback ⓘ v0.5.0

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 v0.5.0 introduces a full vendor dependency system — a hosted registry, `okf.lock` locking, and new `okf pull` / `restore` / `vendor` subcommands — alongside multi-scope BM25 search with layered knowledge precedence.

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

01 Vendor bundle dependency locking and management NEW

95

`okf pull [<bundle-id|git-url>]` installs a vendor knowledge bundle, writes it to `.okf/vendor/<bundle-id>/`, and records it in `okf.lock`; run with no arguments it restores all locked bundles. `okf restore` re-installs and validates all dependencies declared in `okf.lock`, with an optional `--force` flag, enabling reproducible CI/CD environments. `okf vendor list` and `okf vendor remove <bundle-id>` inspect and cleanly remove installed bundles from `.okf/vendor/` and `okf.lock`. The `okf.lock` lockfile (implemented in `pkg/lock`) deterministically tracks vendor packages by source URL, pinned version, SHA-256 checksum, and installation timestamp, and downloaded registry archives are verified against their SHA-256 checksum with automatic rollback on any post-install validation failure.

Pull a scoped vendor knowledge bundle from the OKF Registry into a project and lock it for reproducible restores.

```
$ okf pull @nextjs-15/routing
```

Re-install all locked vendor bundles in a clean CI environment, with `--force` to overwrite any partially written state.

```
$ okf restore --force
```

— Names every subcommand, flag, lockfile field and rollback mechanism v0.5.0

02 Multi-scope BM25 search with layered knowledge precedence NEW

95

A new `--scope <all|project|bundle|vendor|user|system>` flag on `okf search` filters results to a specific memory layer, normalizing BM25 scores and ranking strictly by a 4-tier priority (`project` 100, `vendor` 70, `user` 50, `system` 10). The underlying layering (`pkg/okf/scope.go`) defines canonical storage locations — `./knowledge/` (project), `.okf/vendor/<bundle>/` (vendor), `~/.okf/` (user), and `/etc/okf/` (system) — with deterministic local-shadowing precedence.

Search only the vendor scope to find routing concepts contributed by external bundles, isolating them from local project memory.

```
$ okf search "routing" knowledge --scope vendor
```

— Flag, priority values and paths all given with a runnable example v0.5.0

03 **Hosted registry client for vendor bundles** NEW 80

A new registry client (`pkg/registry`) connects to `https://registry.okf-memory.dev` , configurable via a `--registry <url>` flag or the `OKF_REGISTRY_URL` environment variable, and supports scoped (`@org/bundle`), top-level (`bundle`), and Git archive references (`github.com/org/repo@vX.Y.Z`).

— Names endpoint, flag and env var but no run example v0.5.0

04 **Cross-scope hermetic linking syntax** NEW 65

Adds hermetic cross-scope linking using `@<bundle-id>/<concept-id>.md` for vendor references, `user:<concept-id>.md` for user-scope, and `system:<concept-id>.md` for system-scope, all recognized by `okf validate --strict` without false-positive broken-link errors.

— Syntax named but no worked example of use v0.5.0

v0.5.0 NOTES ↗ LATEST STABLE

OKF Agent Memory v0.5.0 adds a hosted registry, `okf.lock` dependency locking, and multi-scope BM25 search with `--scope` .

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.5.0 https://github.com/okf-memory/okf-agent-memory.git
# already have the repo? check out this version:
$ git checkout v0.5.0
```

└──▷ TRY IT

Pull a scoped vendor knowledge bundle from the OKF Registry into a project and lock it for reproducible restores.

```
$ okf pull @nextjs-15/routing
```

Re-install all locked vendor bundles in a clean CI environment, with `--force` to overwrite any partially written state.

```
$ okf restore --force
```

Search only the vendor scope to find routing concepts contributed by external bundles, isolating them from local project memory.

```
$ okf search "routing" knowledge --scope vendor
```

Adds `okf pull [<bundle-id|git-url>]` subcommand to install a vendor knowledge bundle, write it to `.okf/vendor/<bundle-id>/`, and record it in `okf.lock`; with no arguments it falls back to restoring all locked bundles.

Adds `okf restore` subcommand to re-install and validate all dependencies declared in `okf.lock`, with optional `--force` reinstallation flag, enabling reproducible CI/CD environments.

Adds `okf vendor list` and `okf vendor remove <bundle-id>` subcommands to inspect installed vendor bundles and cleanly remove them from `.okf/vendor/` and `okf.lock`.

Adds `okf.lock` lockfile (implemented in `pkg/lock`) that deterministically tracks vendor packages by source URL, pinned version, SHA-256 checksum, and installation timestamp.

Adds registry client (`pkg/registry`) connecting to `https://registry.okf-memory.dev`, configurable via `--registry <url>` flag or `OKF_REGISTRY_URL` environment variable, supporting scoped (`@org/bundle`) and top-level (`bundle`) slugs plus Git archive references (`github.com/org/repo@vX.Y.Z`).

+4 more

WAS THIS USEFUL? 👍 👎

Docling

Feedback ⓘ v2.132.0 NOTES ↗ RANK

Docling parses PDFs, Office files, and HTML into structured documents with layout, tables, and reading order preserved for downstream RAG pipelines.

> Docling's biggest addition this window is a new FFmpeg-based video ingestion pipeline with per-document timeout handling, alongside authenticated remote image fetching for HTML sources, refined PDF reading-order detection, new PDF widget-annotation fields, and safer TikZ diagram rendering via Tectonic.

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

01 Video pipeline support via FFmpeg with timeout handling

95

BREAKING

Adds video pipeline support for `.mp4` / `.mov` (MP4/QuickTime), `.mkv` / `.webm` (Matroska), and `.avi` (AVI) containers via FFmpeg demuxer, matched strictly by file extension. Adds `--document-timeout` CLI flag to set a per-video time budget in seconds, shared across all FFmpeg calls, defaulting to unset (each call otherwise capped at 5 minutes or video duration, whichever is longer). Adds `PARTIAL_SUCCESS` result status and `TIMEOUT` error recording in `ConversionResult.errors` when a video conversion times out, preserving frames and transcript gathered so far, inspectable via ConversionResult.has_timeout_errors(). Breaking: a video file whose content does not match its extension (e.g. an MKV file named `.mp4`) now fails to open instead of being interpreted as another format.

Cap total processing time for a long video to avoid hanging CI pipelines, while still recovering partial frames and transcript on timeout.

```
$ docling --document-timeout 120 lecture.mp4
```

Check after conversion whether a video hit the timeout so you can requeue or alert on incomplete results.

```python
from docling.document_converter import DocumentConverter

converter = DocumentConverter()
result = converter.convert('lecture.mp4')
if result.has_timeout_errors():
    print('Conversion timed out; partial frames/transcript
saved.')
    print(result.errors)
```

— Names flags, defaults, statuses and a breaking behavior change `product docs`

02 **Authenticated remote image fetching in HTML backend** NEW

95

Adds `fetch_images=True` / `enable_remote_fetch=True` to `HTMLBackendOptions` (CLI: `--html-image-fetch remote`) to opt in to downloading images referenced by an HTML page from remote hosts. Adds a `headers` field on `HTMLBackendOptions` to attach HTTP headers (e.g. `Authorization: Bearer TOKEN`) to remote image downloads, sent only to the source document's origin and dropped on cross-origin redirects. Adds `headers_allowed_origins` to restrict or extend which origins (scheme + host + optional port, e.g. `https://cdn.example.com:8443`) receive the configured headers, replacing the default origin list so the source origin must be re-included if it also needs headers; when `None`, headers are scoped to the source document's own origin (`source_uri`). Adds `--html-image-headers` CLI flag (JSON object) and repeatable `--html-image-headers-origin` flag to pass per-origin download headers from the command line.

Fetch remote images from a gated HTML page, sending a Bearer token only to the source domain and a CDN.

```python
from docling.datamodel.pipeline_options import HTMLFormatOption
from docling.backend.html_backend import HTMLBackendOptions
from docling.document_converter import DocumentConverter

html_opts = HTMLBackendOptions(
    fetch_images=True,
    enable_remote_fetch=True,
    headers={"Authorization": "Bearer TOKEN"},
    headers_allowed_origins=[
        "https://example.com",
        "https://cdn.example.com",
    ],
)
converter = DocumentConverter(
    format_options={"html":
HTMLFormatOption(html_options=html_opts)}
)
result = converter.convert("https://example.com/page.html")
```

Fetch remote HTML images with auth headers from the CLI, restricting header forwarding to the source host and a CDN.

```
$ uv run docling --to dclx --html-image-fetch remote --html-image-headers '{"Authorization": "Bearer TOKEN"}' --html-image-headers-origin https://example.com --html-image-headers-origin https://cdn.example.com --output-file ./page.dclx https://example.com/page.html
```

— Full mechanism, defaults and runnable code/CLI examples given `product docs`

---

03 **Reading-order algorithm improvements for PDF structure** IMPROVED

75

Adds `use_reading_order_separators` option (and `--no-reading-order-separators` CLI flag) letting the rule-based reading-order stage use visible PDF vector rules as structural signals when separating columns or horizontal bands. Also updates the underlying reading-order algorithm for PDF documents to improve structural extraction accuracy generally.

Convert a multi-column PDF twice — once with separator-assisted reading order and once without — to diff the structural difference.

```
$ uv run docling --to dclx --output-file ./Elsevier-with-separators.dclx ./Elsevier.pdf
uv run docling --to dclx --no-reading-order-separators --output-file ./Elsevier-without-separators.dclx ./Elsevier.pdf
```

— Names the flag and mechanism but the general algorithm update is vague v2.132.0

04 **TikZ diagram rendering via Tectonic engine** NEW 70

Adds support for rendering TikZ diagrams to images via the `'tectonic'` engine, with `--untrusted` and `--only-cached` defaults for safer handling of untrusted LaTeX; diagrams referencing absolute or parent-directory paths are kept as TikZ code rather than rendered. Adds a shell-escape opt-in for Tectonic TikZ rendering, disabled by default for safer processing of untrusted LaTeX.

— Explains defaults and safety behavior but no runnable example product docs

05 **`--output-file` CLI flag for exact output path** NEW 60

Adds `--output-file` CLI argument to select an exact destination path when converting a single input to a single output format.

— Names exact flag but minimal explanation of behavior product docs

THINNER COVERAGE BELOW

06 **Widget annotation state fields for PDF forms** NEW 55

Adds `widget_field_flags` and `widget_appearance_state` fields on PDF widget annotation objects to expose raw AcroForm field state (`/Ff` mask and `/AS` name) from PDF documents.

— Names fields precisely but no usage example provided product docs

└──▷ BREAKING ON UPGRADE

! A video file whose content does not match its extension (e.g. an MKV file named `.mp4`) now fails to open instead of being interpreted as another format.

v2.132.0 NOTES ↗ LATEST STABLE

Docling v2.132.0 updates the PDF reading-order algorithm for improved document structure extraction.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.132.0 https://github.com/docling-project/docling.git
# already have the repo? check out this version:
$ git checkout v2.132.0
```

Updates the reading-order algorithm for PDF documents to improve structural extraction accuracy.

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ snapshot-20261001 NOTES ↗

RANK

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

Adds `/v2/threads/stats` API, SSE thread streaming, `ls_provider` for Gateway trace metadata, `LICENSE_EXPIRATION_WARNING_DAYS` for self-hosted operators, Gateway header-timeout fallback, OAuth `owner_type` / `owner_id` params, and trajectory gRPC isolation.

Adds `GET /v2/threads/stats` endpoint to retrieve thread and trace counts, latency, token, cost, and feedback statistics for a tracing project, including on self-hosted ClickHouse-backed deployments.

Adds `LICENSE_EXPIRATION_WARNING_DAYS` config option for self-hosted operators to control how many days before license expiration the warning banner appears (default: 7 days).

Adds `ls_provider` field on OpenAI-compatible model configurations to populate trace metadata and match custom models to provider-specific pricing through the LLM Gateway.

OAuth authorization sessions now accept `owner_type` and `owner_id` parameters, allowing a human to authorize a managed credential for either a user or an agent.

OAuth authorization sessions now accept `workspace` owners, allowing one managed OAuth credential to be shared across a workspace.

+11 more

NEW INTEGRATION Claude Fable 5.1 model support in Playground

NEW INTEGRATION LangSmith MCP connector Client ID Metadata Document support

NEW INTEGRATION ECR private registry IAM role assumption for Sandboxes

UPDATED INTEGRATION Amazon Bedrock Playground timeout and retry parity

NEW INTEGRATION IssueBench Opus 5 benchmark task

└──▷ BREAKING ON UPGRADE

! Dataset CSV and JSONL exports now require both dataset read AND download permissions — API callers that previously had only read permission will receive a forbidden response on export requests.

WAS THIS USEFUL? 👍 👎

Arize Phoenix

Feedback ⓘ `@arizeai/phoenix-client@7.16.0`

NOTES ↗ RANK

Arize Phoenix is an open-source platform that monitors LLM application traces and evaluates their outputs.

> Arize Phoenix added project-level annotation config management, inline eval results in the trace tree, GraphQL experiment sorting/filtering, and automatic error marking for failed bash tool spans.

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

01 **GraphQL sorting and filtering for experiments** IMPROVED

78

Adds `sort` and `sequenceNumbers` arguments to the `Dataset.experiments` GraphQL field, enabling ordered and filtered experiment retrieval via `sequence_number` and `sort_dir`.

— Names exact GraphQL field and arguments `arize-phoenix-v20.17.0`

02 **Project annotation config management helpers** NEW

73

Adds `listProjectAnnotationConfigs`, `assignProjectAnnotationConfig`, `unassignProjectAnnotationConfig`, and `setProjectAnnotationConfigs` helpers to the `projects` subpath of the client, for managing which annotation configs are associated with a project.

— Names exact client helper functions but no usage example `@arizeai/phoenix-client@7.16.0`

THINNER COVERAGE BELOW

03 **Inline eval results in trace tree** NEW

45

Displays evaluation results inline within the trace tree view, letting practitioners see eval scores alongside each span without leaving the tracing UI.

— Describes behavior but no exact UI path or config `arize-phoenix-v20.18.0`

04 **Automatic error marking for bash tool spans** IMPROVED

45

Marks bash tool spans as errors automatically when the subprocess exits with a non-zero exit code.

— Clear mechanism but no API or config surface named `arize-phoenix-v20.17.0`

`@arizeai/phoenix-client@7.16.0` NOTES ↗ LATEST STABLE

Adds four project annotation config helpers — `listProjectAnnotationConfigs`, `assignProjectAnnotationConfig`, `unassignProjectAnnotationConfig`, and setProjectAnnotationCon

└──▷ GET THIS VERSION ▾

```
$ git clone --branch @arizeai/phoenix-client@7.16.0
https://github.com/Arize-ai/phoenix.git
# already have the repo? check out this version:
$ git checkout @arizeai/phoenix-client@7.16.0
```

arize-phoenix-v20.18.0 NOTES ↗ STABLE

Arize Phoenix v20.18.0 surfaces eval results directly in the trace tree UI.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch arize-phoenix-v20.18.0
https://github.com/Arize-ai/phoenix.git
# already have the repo? check out this version:
$ git checkout arize-phoenix-v20.18.0
```

Displays evaluation results inline within the trace tree view, letting practitioners see eval scores alongside each span without leaving the tracing UI.

arize-phoenix-v20.17.0 NOTES ↗ STABLE

Phoenix adds experiment sorting/filtering via `sequence_number` and `sort_dir` in GraphQL, plus bash tool span error marking for agents.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch arize-phoenix-v20.17.0
https://github.com/Arize-ai/phoenix.git
# already have the repo? check out this version:
$ git checkout arize-phoenix-v20.17.0
```

Adds `sort` and `sequenceNumbers` arguments to the `Dataset.experiments` GraphQL field for ordered and filtered experiment retrieval.

Marks bash tool spans as errors automatically when the subprocess exits with a non-zero exit code.

WAS THIS USEFUL? 👍 👎

Comet Opik

Feedback ⓘ 2.2.87 NOTES ↗ RANK

Opik traces, evaluates, and monitors LLM applications and agentic workflows, with datasets, an LLM-as-judge metric library, and production dashboards.

> Opik 2.2.87 adds a simpler MCP setup command, new session-analysis and experiment-comparison fields, and restores a kill-switch for Agent Insights.

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

01 **npx wrapper for MCP setup** NEW 65

Adds an `npx opik` wrapper for the `uvx` MCP setup command, giving Node.js users a one-line entry point to configure the Model Context Protocol integration.

– Names exact command but no further mechanism detail 2.2.87

THINNER COVERAGE BELOW

02 **Agent Insights kill-switch restored** IMPROVED 55

Restores `TOGGLE_AGENT_INSIGHTS_ENABLED` as a kill-switch environment variable to enable or disable the Agent Insights feature.

– Exact env var named, minimal behavioural detail 2.2.87

03 **Task outcome field in session analysis** NEW 50

Adds `tasks.outcome` field to `cipx_session_analysis`, exposing per-task outcome data in session analytics.

– Names field and table but no usage context 2.2.87

04 **Prompt version in experiment compare view** NEW 40

Shows each experiment's prompt version in the compare view, making prompt-version differences visible across experiment runs.

– UI location given but no further mechanism 2.2.87

2.2.87 NOTES ↗ LATEST STABLE

Opik 2.2.87 adds an `npx opik` wrapper for uvx MCP setup, a prompt-version column in experiment compare view, and `tasks.outcome` in session analysis.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch 2.2.87 https://github.com/comet-ml/opik.git
# already have the repo? check out this version:
$ git checkout 2.2.87
```

Adds `npx opik` wrapper for the `uvx` MCP setup command, giving Node.js users a one-line entry point to configure the Model Context Protocol integration.

Adds `tasks.outcome` field to `cipx_session_analysis`, exposing per-task outcome data in session analytics.

Restores `TOGGLE_AGENT_INSIGHTS_ENABLED` as a kill-switch environment variable to enable or disable the Agent Insights feature.

Shows each experiment's prompt version in the compare view, making prompt-version differences visible across experiment runs.

WAS THIS USEFUL? 👍 👎

Langfuse

Feedback ⓘ v4.49.0 NOTES ↗ RANK

Langfuse provides tracing, evaluation, and monitoring for LLM applications.

> Langfuse v4.49.0 adds the ability to filter sessions by tool usage in the session table.

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

01 **Tool filters in session table** NEW 40

The session table now supports tool filters, enabling filtering sessions by tool usage.

— Names the UI surface but no mechanism or config details. v4.49.0

> v4.49.0 NOTES ↗ LATEST STABLE
>
> Langfuse v4.49.0 adds tool filters to the session table and an org-level analytics view.
>
> └──▷ GET THIS VERSION ▾
>
> ```
> $ git clone --branch v4.49.0
> https://github.com/langfuse/langfuse.git
> # already have the repo? check out this version:
> $ git checkout v4.49.0
> ```
>
> Adds tool filters to the session table, enabling filtering sessions by tool usage.

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Open-Weight Models

DeepSeek V4 Pro 0813 Feedback ⓘ

price increase · Sep 30, 2026 MODEL ↗

DeepSeek V4 Pro 0813 is a chat model from DeepSeek.

Price increase on OpenRouter (deepseek/deepseek-v4-pro-0813): $0.3458 → $1.32 in, $1.0375 → $3.96 out per million tokens (+282%), first seen 30 Sep 2026.

WAS THIS USEFUL? 👍 👎

Z.ai GLM 5.3 Flash Feedback ⓘ context change · Sep 30, 2026

MODEL ↗

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

Context window shrank from 1311K to 1049K tokens on OpenRouter (z-ai/glm-5.3-flash), first seen 30 Sep 2026.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · October 1, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue