THE AI TOOLCHAIN

NO. 036

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED SEPTEMBER 24, 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 308 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 48 tools

▾ // THE BRIEF HIDE SAME ISSUE, SAME PROMPT, TWO WRITERS: GPT-5.5 | Opus 5

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

Three unrelated things stand out today: NVIDIA Model Optimizer makes FP8 quantization less bespoke; MCP TypeScript SDK tightens OAuth

vim ? help

boundaries around individual MCP primitives; Braintrust turns trace review into automated pattern-finding instead of manual spelunking.

DEPLOY

Quantize expert-heavy models with less custom calibration work

Per-expert weight quantization for Transformer Engine MoE experts, NVFP4 activation-headroom calibration, MLflow tracking, and new sensitivity scoring give teams more built-in machinery for post-training optimization. That reduces the hand-tuned recipe work usually needed before large or sparse models can be served within latency and memory budgets.

NVIDIA Model Optimizer

GOVERN

Bind MCP access to the caller and the primitive being invoked

The MCP TypeScript SDK added DPoP sender-constrained OAuth tokens and per-primitive scope enforcement, while gram added workload JWT-bearer grants, expanded agent scopes, and upstream-account bindings. Together they make agent/tool access less dependent on broad shared credentials, so a stolen token or over-permissioned agent has a smaller blast radius.

MCP TypeScript SDK · gram

EVALUATE

Find repeated agent failures without reading every trace

Braintrust added automated trace analysis with Pattern frequency charts and the Debugger. Instead of sampling runs by hand and guessing which failures are systemic, teams can surface recurring behaviors and prioritize the regressions that affect many executions.

Braintrust

BUILD

Let agent tools hand off long jobs instead of blocking the run

Mastra tools can now delegate long-running work to the platform for background tracking, alongside typed workflow steps and a Classifier primitive for AI-based evaluation. That lets an agent start durable work and continue orchestration without keeping the whole interaction open just to wait on an external task.

Mastra

Does **GPT-5.5** read better? 👍 👎

BUILD

◆ AI Coding Agents

Replit Agent

Feedback ⓘ snapshot-20260924 NOTES ↗ RANK

Replit Agent is an AI coding agent that builds, tests, and deploys applications in Replit.

> Replit Agent added a documentation discovery endpoint for programmatic access to its docs index.

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

01 **Documentation index endpoint at /llms.txt** NEW 60

Adds a `/llms.txt` endpoint that returns the complete documentation index, enabling programmatic discovery of all available documentation pages.

— Names exact endpoint but no further mechanism detail. snapshot-20260924

> snapshot-20260924 NOTES ↗ LATEST STABLE
>
> Replit Agent exposes a documentation index at `/llms.txt` for programmatic discovery of all available pages.

WAS THIS USEFUL? 👍 👎

Vercel v0

Feedback ⓘ snapshot-20260924 NOTES ↗ RANK

The v0 generative UI platform builds web applications from natural-language prompts.

> v0 added new MCP server tools for chat automation, an Enterprise-only AI Gateway model picker, and workspace controls for chat visibility, team design systems, and project activity visibility.

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

01 **Enterprise AI Gateway model picker** NEW 85

Enterprise team owners can enable the Allow custom models toggle in Workspace settings, letting members browse and add AI Gateway models via Add models in the chat composer, with per-model token pricing review and Thinking/Fast Mode customization where supported.

Allow Enterprise team members to use AI Gateway models beyond the default v0 models, including future releases.

📍 In v0, go to Workspace settings and enable Allow custom models.

– Exact settings path and toggle name with runnable steps snapshot-20260924

02 **New MCP server tools for chat automation** NEW 75

The v0 MCP server adds `listMessages`, which reads a chat's history, and `resolveTask`, which answers plan approvals, questions, permission requests, and integration prompts. The server also now returns scope-aware chat URLs.

– Names exact tool functions but no invocation example given snapshot-20260924

03 **Default chat visibility setting for workspaces** NEW 75

A new Default Chat Visibility setting in Settings → Workspace lets teams choose Private, Team can view, or Team can edit to control visibility of new team-workspace chats.

Make all new team chats visible to teammates by default so collaboration starts immediately without manual sharing.

📍 In v0, go to Settings → Workspace → Default Chat Visibility and select Team can view or Team can edit.

– Named setting with exact options and navigation path snapshot-20260924

04 **Team-default design systems** NEW 60

Teams can now set a team-default Design System, including Design Systems 2.0 skills and built-in examples, and choose team or personal scope when saving a design system from the save card.

— Names save card and scope option but no exact path given `snapshot-20260924`

THINNER COVERAGE BELOW

05 **Teammate activity avatars on project cards** NEW 45

Project cards now display avatars of teammates active in visible project chats within the last 24 hours, ordered by most recent activity.

— Describes the change but no way to configure or navigate to it `snapshot-20260924`

`snapshot-20260924` NOTES ↗ LATEST STABLE

v0 gains Enterprise model picker with AI Gateway access, team-default chat visibility, and new MCP server tools.

└──▷ **HOW TO FIND IT**

Make all new team chats visible to teammates by default so collaboration starts immediately without manual sharing.

📍 In v0, go to Settings → Workspace → Default Chat Visibility and select `Team can view` or `Team can edit`.

Allow Enterprise team members to use AI Gateway models beyond the default v0 models, including future releases.

📍 In v0, go to Workspace settings and enable `Allow custom models`.

Adds `listMessages` and `resolveTask` tools to the v0 MCP server — `listMessages` reads a chat's history and `resolveTask` answers plan approvals, questions, permission requests, and integration prompts; the server now also returns scope-aware chat URLs.

Adds `Allow custom models` toggle in Workspace settings for Enterprise team owners, enabling browsing and adding AI Gateway models via `Add models` in the chat composer, with per-model token pricing review and Thinking/Fast Mode customization where supported.

Adds configurable `Default Chat Visibility` setting in Settings → Workspace, with options `Private`, `Team can view`, and `Team can edit`, controlling visibility of new team-workspace chats.

Project cards now display avatars of teammates active in visible project chats within the last 24 hours, ordered by most recent activity.

> Enables setting a team-default Design System (including Design Systems 2.0 skills and built-in examples) and choosing team or personal scope when saving a design system from the save card.

WAS THIS USEFUL?  

Anysphere Cursor

Feedback ⓘ snapshot-20260924 NOTES ↗

RANK

Built to make you extraordinarily productive, agents turn ideas into code. Accelerate development by handing off tasks to Cursor.

> Cursor shipped two new automation bots for Teams and Enterprise — Rollouts for deploy health monitoring and Security Review for automated PR vulnerability scanning — alongside per-automation service accounts for isolating automation identity from personal auth.

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

01 Security Review automated PR security analysis NEW

85

Enabled from the dashboard's automations tab, Security Review posts one review comment per pull request reporting exploitable bugs — covering SQL/command/template injection, auth bypasses, committed secrets and credentials, SSRF, unvalidated redirects, unsafe deserialization, and dependency changes introducing known vulnerabilities — with severity, attack path, and a proposed fix per finding. It supports team rules to enforce codebase-specific policies (e.g. which external call paths are permitted, which tables must not be queried from a request handler) on every PR, and automatically skips draft PRs.

Enable Security Review on a repository so every merged PR is automatically checked for exploitable bugs before code ships.

> 📍 In the Cursor dashboard, go to the automations tab, select Security Review, choose the repositories you want reviewed, and save. Security Review will post a review comment with severity, attack path, and proposed fix on each subsequent non-draft pull request.

— Names vulnerability classes and policy examples, UI setup only. snapshot-20260924

02 Rollouts deploy health monitoring bot NEW

83

Enabled from the dashboard's automations tab, Rollouts attaches a monitor to every pull request, writes a monitoring plan as a PR comment listing identified risks, intended effects, signals to check, and instrumentation gaps, then tracks deploy health per environment — verified healthy, regression detected, or inconclusive — against logs, metrics, and traces. It integrates with Origin or GitHub for source control, continuous delivery systems for deploy events, and Datadog and other telemetry providers for signals.

Enable Rollouts to monitor a new deployment and catch regressions per environment as soon as the next PR is opened.

> 📍 In the Cursor dashboard, go to the automations tab, select `Rollouts`, connect your source control (Origin or GitHub), your continuous delivery system, and your telemetry provider (e.g. Datadog), then save. Rollouts will generate a monitoring plan as a PR comment on the next pull request and track deploy health across staging and production.

— Clear mechanism and integrations named, only UI setup given. `snapshot-20260924`

└──▷ ALSO FROM THESE RELEASES

.
.
.

└──▷ BREAKING ON UPGRADE

! Switching an automation's `Run as` to `Service account` stops using your personal auth; any webhook triggers require the webhook API key to be regenerated, and any MCPs or OAuth integrations must be reconfigured for the team service account.

! An automation set to `Run as: Service account` stops running if its `automation-<id>` service account is archived or deleted.

`snapshot-20260924` NOTES ↗ **LATEST** STABLE

Cursor launches Rollouts (deploy health monitoring) and Security Review (automated PR security analysis) bots for Teams and Enterprise.

└──▷ HOW TO FIND IT

Enable Security Review on a repository so every merged PR is automatically checked for exploitable bugs before code ships.

> 📍 In the Cursor dashboard, go to the automations tab, select `Security Review`, choose the repositories you want reviewed, and save. Security Review will post a review comment with severity, attack path, and proposed fix on each subsequent non-draft pull request.

Enable Rollouts to monitor a new deployment and catch regressions per environment as soon as the next PR is opened.

> 📍 In the Cursor dashboard, go to the automations tab, select `Rollouts`, connect your source control (Origin or GitHub), your continuous delivery system, and your telemetry provider (e.g. Datadog), then save. Rollouts will generate a monitoring plan as a PR comment on the next pull request and track deploy health across staging and production.

Adds Rollouts bot, enabled from the dashboard's automations tab, which attaches a monitor to every pull request, writes a monitoring plan as a PR comment listing identified risks, intended effects, signals to check, and instrumentation gaps, then tracks deploy health per environment (verified healthy, regression detected, or inconclusive) against logs, metrics, and traces.

Adds Security Review bot, enabled from the dashboard's automations tab, which posts one review comment per pull request reporting exploitable bugs — covering SQL/command/template injection, auth bypasses, committed secrets and credentials, SSRF, unvalidated redirects, unsafe deserialization, and dependency changes introducing known vulnerabilities — with severity, attack path, and a proposed fix per finding.

Supports team rules in Security Review to enforce codebase-specific policies (e.g. which external call paths are permitted, which tables must not be queried from a request handler) on every PR.

Rollouts integrates with Origin or GitHub for source control, continuous delivery systems for deploy events, and Datadog and other telemetry providers for signals.

Security Review skips draft PRs automatically.

WAS THIS USEFUL? 👍 👎

Cline

Feedback ⓘ desktop-v0.0.35 NOTES ↗ CODE ↗ RANK

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

> Cline shipped Linux support for its desktop app, a new `ai&` provider spanning desktop, CLI and SDK, provider-backed live voice transcription, and shared max-tokens crash-recovery and plugin-slash-command infrastructure built once in the SDK and surfaced across desktop and CLI.

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

01 New ai& provider (AIAND_API_KEY) NEW 85

Adds a new `ai&` provider, an OpenAI-compatible endpoint serving open-weight models from Japan, enabled via the `AIAND_API_KEY` environment variable with GLM 5.3 as the default model; exposed in the SDK as the `aiand` provider and available in the desktop app's provider list.

Use the new ai& provider with a Japanese open-weight model by setting your API key before running a session.

```
$ AIAND_API_KEY=your_key_here cline 'Refactor this module for readability'
```

— Names the env var, endpoint origin and default model with a runnable example

desktop-v0.0.35 cli-v3.0.65 sdk/sdk/v0.0.86

02 Provider-backed live voice transcription NEW 85

Voice input now supports provider-backed live streaming transcription via OpenAI, Vercel AI Gateway, and ElevenLabs, with automatic fallback to the browser's recognizer when the network drops mid-recording. The SDK exposes this via `getBuiltinStreamingTranscriptionModels` in `@cline/llms`, covering native OpenAI (`gpt-realtime-whisper` with a transcription-bound client secret), Vercel AI Gateway, and ElevenLabs; batch transcription via the AI SDK adds retries, cancellation, `providerOptions`, audio-format auto-detection, and returns `segments` and `warnings` alongside the transcript text.

— Names the SDK function, providers and returned fields sdk/sdk/v0.0.86

desktop-v0.0.35

03 Plugin slash commands and retry handling IMPROVED 80

Plugin slash commands (e.g. `/goal`) now execute the plugin's registered handler in the desktop app instead of being sent as plain text, with enabled commands surfaced in the slash menu. The SDK adds `createPluginCommandService` in `@cline/core` as a

shared core service for plugin slash-command discovery, loading, name/result normalization, and execution, used by both desktop and CLI. Failed plugin loads are cached and retried after 30 seconds instead of spawning a sandbox on every prompt, so transient sandbox timeouts self-recover.

— Names the shared service, example command and retry timing `sdk/sdk/v0.0.86` `cli-v3.0.65` `desktop-v0.0.35`

---

04 **Output-token-limit crash recovery for local models** IMPROVED

80

Long replies on local models (llama.cpp, Ollama, LM Studio) that hit the output-token limit now compact the conversation and retry once instead of failing the run, keeping any partial answer as a fallback. The SDK adds a `task.max_tokens_recovery` lifecycle event (values `started`, `retried`, `failed`) to measure recovery effectiveness, and routes text-only truncated turns through the same `prepareTurn` forced-compaction path used for context-window overflow, falling back to the existing nudge-and-retry loop.

— Names the lifecycle event, its values and internal recovery path `sdk/sdk/v0.0.86` `cli-v3.0.65` `desktop-v0.0.35`

---

05 **CLI session history persistence** IMPROVED

70

Persists error messages in session history so they reappear on resume without being sent to the model or counted by compaction. Persists `cline history update --title` and `--prompt` changes for hub-managed sessions so renames survive restarts and no longer drop other keys such as pinned state.

— Names the exact CLI flags and persistence behavior `cli-v3.0.65`

---

06 **Model list error surfacing for endpoint-owned providers**

IMPROVED

60

Model lists for LiteLLM, Baseten, Hicap, Poolside, Ollama, and LM Studio now show the actual error when an endpoint is unreachable instead of returning an empty list — the SDK propagates errors from `getLocalProviderModels` so hosts can surface the real cause — and the desktop app trusts the OS certificate store for corporate CA-signed endpoints.

— Names six providers and the underlying function but no user action `sdk/sdk/v0.0.86` `desktop-v0.0.35`

---

07 **Linux desktop support** NEW

60

Adds Linux support to Cline Desktop, shipping x64 `.deb` and `.rpm` packages with a native GTK folder picker and background update installation via the Restart now action.

— Names packages and update mechanism but no install command `desktop-v0.0.35`

---

08 **Diagnostics export in Settings** NEW 60

Adds a Diagnostics row in Settings with an Export… action that writes a redacted text file (app version, OS, settings, recent sidecar and hub logs, selected session manifests) to your Downloads folder, safe for attaching to GitHub issues.

— Names UI location and file contents, a clear starting point desktop-v0.0.35

09 **Yolo mode output tightening** IMPROVED 60

Tightens yolo mode (`-y`) output rules: shorter plans, no preamble before routine tool calls, and code/edits written directly into tool calls instead of drafted in text first.

Run fully autonomous tasks with yolo mode to cut narration overhead and get edits written straight into tool calls.

```
$ cline -y 'Add input validation to all API handlers and write unit tests'
```

— Named flag with a runnable example command cli-v3.0.65

10 **./cloud subpath export in @cline/core** NEW 60

Adds a `./cloud` subpath export to `@cline/core` exposing the shared cloud session runtime (API client, controller, snapshots, state) for host consumption.

— Names the exact export path and exposed components sdk/sdk/v0.0.86

THINNER COVERAGE BELOW

11 **Model catalog refresh to 6,386 models** IMPROVED 55

Refreshed the model catalog from 6,237 to 6,386 models: adds `cline-free/mimo-v2.6-flash`, drops `z-ai/glm-5.3-flash`, `cline-free/solar-pro4`, and `poolside/laguna-s-2.1:free`, and updates default models for 19 providers, with 11 (including GitHub Copilot and Vertex) now defaulting to Claude Opus 5.5.

— Exact numbers and model names but nothing for the reader to do sdk/sdk/v0.0.86 desktop-v0.0.35

12 **toPosixSeparators helper export** IMPROVED 45

Exports `toPosixSeparators` from `@cline/shared`, replacing four separate copies of the helper across core and the VS Code extension.

— Names the export but is a minor internal refactor sdk/sdk/v0.0.86

13 **Per-provider reasoning effort persistence** IMPROVED 25

Reasoning effort choice is now remembered per provider across session reopens and provider switches.

— States behaviour only, no surface or mechanism named `desktop-v0.0.35`

`desktop-v0.0.35` NOTES ↗ CODE ↗ **LATEST** STABLE

Cline Desktop adds Linux support, plugin slash commands, diagnostics export, live voice transcription, and a new ai& provider.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch desktop-v0.0.35
https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout desktop-v0.0.35
```

Adds Linux support to Cline Desktop, shipping x64 `.deb` and `.rpm` packages with native GTK folder picker and background update installation via Restart now.

Adds Diagnostics row in Settings with an Export… action that writes a redacted text file (app version, OS, settings, recent sidecar and hub logs, selected session manifests) to your Downloads folder, safe for attaching to GitHub issues.

Plugin slash commands (e.g. `/goal`) now execute the plugin's registered handler in the desktop app instead of being sent as plain text, with enabled commands surfaced in the slash menu.

Voice input now supports provider-backed live streaming transcription via OpenAI, Vercel AI Gateway, and ElevenLabs, with automatic fallback to the browser's recognizer when the network drops mid-recording.

Long replies on local models (llama.cpp, Ollama, LM Studio) that hit the output-token limit now compact the conversation and retry once instead of failing the run.

+3 more

`cli-v3.0.65` NOTES ↗ CODE ↗ STABLE

Cline CLI v3.0.65 adds ai& provider with `AIAND_API_KEY`, tighter yolo mode output rules, and max-tokens crash recovery for local models.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch cli-v3.0.65 https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout cli-v3.0.65
```

└──▷ **TRY IT**

Use the new ai& provider with a Japanese open-weight model by setting your API key before running a session.

```
$ AIAND_API_KEY=your_key_here cline 'Refactor this module for readability'
```

Run fully autonomous tasks with yolo mode to cut narration overhead and get edits written straight into tool calls.

```
$ cline -y 'Add input validation to all API handlers and write unit tests'
```

Adds `AIAND_API_KEY` environment variable to enable the new ai& provider, an OpenAI-compatible endpoint serving open-weight models from Japan; default model is GLM 5.3.

Tightens yolo mode (`-y`) output rules: shorter plans, no preamble before routine tool calls, and code/edits written directly into tool calls instead of drafted in text first.

Automatically compacts the conversation and retries once when a local model (llama.cpp, Ollama, LM Studio) hits the output-token limit mid-answer, keeping any partial answer as a fallback.

Persists error messages in session history so they reappear on resume without being sent to the model or counted by compaction.

Persists `cline history update --title` and `--prompt` changes for hub-managed sessions so renames survive restarts and no longer drop other keys such as pinned state.

+1 more

sdk/sdk/v0.0.86 NOTES ↗ CODE ↗ STABLE

Cline SDK v0.0.86 adds a new `aiand` provider, streaming transcription across three engines, a `./cloud` subpath export, and a new `task.max_tokens_recovery` lifecycle event.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch sdk/sdk/v0.0.86
https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout sdk/sdk/v0.0.86
```

Adds `getBuiltinStreamingTranscriptionModels` in `@cline/llms`, covering native OpenAI (`gpt-realtime-whisper` with a transcription-bound client secret), Vercel AI Gateway, and ElevenLabs; batch transcription via AI SDK adds retries, cancellation, `providerOptions`, audio-format auto-detection, and returns `segments` and `warnings` alongside the transcript text.

Adds a `./cloud` subpath export to `@cline/core` exposing the shared cloud session runtime (API client, controller, snapshots, state) for host consumption.

Adds `createPluginCommandService` in `@cline/core` as a shared core service for plugin slash-command discovery, loading, name and result normalization, and execution; failed plugin loads are cached and retried after 30 seconds instead of spawning a sandbox on every prompt.

Adds a `task.max_tokens_recovery` lifecycle event (values: `started`, `retried`, `failed`) to measure how often compact-and-retry recovery helps on output-token-limit truncation.

Adds compact-and-retry recovery for text-only turns truncated at the output-token limit, routing through the same `prepareTurn` forced-compaction path used for context-window overflow before falling back to the existing nudge-and-retry loop.

+3 more

WAS THIS USEFUL? 👍 👎

Letta Code

Feedback ⓘ v0.33.0 NOTES ↗ RANK

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

> Letta Code's window focused on surfacing agent workflow activity in real time, both through a new protocol-level exposure and a live status display in the CLI.

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

01 **Live workflow status rows in CLI** NEW 45

The CLI now displays live workflow status rows beneath the input prompt, giving real-time visibility into agent activity as it runs.

— Describes UI location but no exact command v0.32.19

02 **Workflow state exposed via protocol layer** NEW 30

Running agent workflows are now exposed through the protocol layer, making in-flight workflow state visible to consumers beyond the CLI itself.

— Names the layer but no endpoint or schema given v0.33.0

v0.33.0 NOTES ↗ LATEST STABLE

Letta Code v0.33.0 offloads Git conflict repair and memory upkeep to a background worker and exposes running workflows via protocol.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.33.0 https://github.com/letta-ai/letta-code.git
# already have the repo? check out this version:
$ git checkout v0.33.0
```

Exposes running workflows through the protocol layer, making in-flight agent workflow state visible.

v0.32.19 NOTES ↗ STABLE

Letta Code v0.32.19 adds live workflow status rows displayed beneath the CLI input prompt.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.32.19 https://github.com/letta-ai/letta-code.git
# already have the repo? check out this version:
$ git checkout v0.32.19
```

Displays live workflow status rows beneath the input line in the CLI, giving real-time visibility into agent activity as it runs.

WAS THIS USEFUL? 👍 👎

OpenAI Codex CLI

Feedback ⓘ rust-v0.158.0-alpha.5 NOTES ↗

CODE ↗ RANK

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

> OpenAI Codex CLI's latest alpha releases focus on securing the exec-server with WebSocket authentication, extending Guardian authorization coverage, and rounding out observability, TUI diagnostics, and retry handling across the agent.

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

01 **WebSocket authentication for exec-server** NEW 95

Adds `--ws-auth` options to `codex exec-server` for WebSocket listener authentication, supporting capability tokens configured by token file or SHA-256 digest and signed JWT bearer tokens. Validates `Authorization: Bearer TOKEN` before each WebSocket upgrade, rejecting missing or invalid credentials with HTTP 401.

Secure a local exec-server WebSocket listener with JWT bearer token authentication so only authorized clients can connect.

```
$ codex exec-server --ws-auth <token-or-config>
```

— Full mechanism, flag, header and status code all given. rust-v0.158.0-alpha.5

02 **Observability and metrics improvements** NEW 65

Adds opt-in OTLP logging for final agent responses, exposes tool dispatch and timing observations to extensions, adds bounded buffering for global operation metrics, and records the observed sandbox backend as a `sandbox_backend` metric tag in command execution analytics, distinguishing unknown/missing backends from explicitly unsandboxed processes.

— Names OTLP and sandbox_backend tag but little mechanism. rust-v0.158.0-alpha.5 rust-v0.158.0-alpha.3

03 **Mermaid rendering fallback notice in TUI** IMPROVED 65

Shows a dimmed notice in the TUI above preserved Mermaid source when a diagram uses unsupported syntax, exceeds renderer size limits, or does not fit the terminal width, instead of silently falling back.

— Explains trigger conditions and behavior clearly. rust-v0.158.0-alpha.3

THINNER COVERAGE BELOW

04 **Guardian authorization coverage extended** IMPROVED 50

Guardian authorization now recognizes `user_message` tools in its authorization context, and applies Guardian computer-use review to the Browser connector.

— Names the tool type and connector but no mechanism. rust-v0.158.0-alpha.5

05 **Retry-After header handling** IMPROVED 40

Honors Retry-After headers and preserves server retry deadlines when retrying requests.

— Names header but no further mechanism given. rust-v0.158.0-alpha.5

06 **Faster startup via WebSocket preconnect overlap** IMPROVED 30

Overlaps startup WebSocket preconnect with tool discovery for faster initialization.

— Describes mechanism briefly, no metrics given. rust-v0.158.0-alpha.5

07 **Restricted default Bedrock GovCloud model catalog** BREAKING 30

Restricts the default Bedrock GovCloud model catalog.

— Names GovCloud catalog but no restriction specifics. rust-v0.158.0-alpha.5

08 **Configurable copy-on-select for transcript selections** NEW 30

Adds configurable copy-on-select for transcript selections.

— Names UI behavior, lacks config key details. rust-v0.158.0-alpha.5

09 **Bearer tokens for app-server executor connections** NEW 25

Supports bearer tokens for app-server executor connections.

— Only a one-line mention, no config detail. rust-v0.158.0-alpha.5

10 **Model-specific tool description prefixes** IMPROVED 25

Adds model-specific prefixes to indirect tool descriptions.

— Single line, no example of mechanism. rust-v0.158.0-alpha.5

11 **Rollout history attached to diagnostics** IMPROVED 25

Attaches inherited rollout history to diagnostic reports.

— Brief mention, no format detail. rust-v0.158.0-alpha.5

12 **History and notes without experimental context capability** IMPROVED 25

Allows history and notes without experimental context capability.

— Terse description, no mechanism given. rust-v0.158.0-alpha.5

13 **Portable project trust lookup APIs** NEW 25

Adds portable project trust lookup APIs.

— No API names or usage given. rust-v0.158.0-alpha.5

rust-v0.158.0-alpha.5 NOTES ↗ CODE ↗ PRE-RELEASE

Codex exec-server gains opt-in WebSocket authentication with bearer tokens and capability tokens, plus OTLP logging for agent responses.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch rust-v0.158.0-alpha.5
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.158.0-alpha.5
```

└──▷ TRY IT

Secure a local exec-server WebSocket listener with JWT bearer token authentication so only authorized clients can connect.

```
$ codex exec-server --ws-auth <token-or-config>
```

Adds `--ws-auth` options to `codex exec-server` for WebSocket listener authentication, supporting capability tokens configured by token file or SHA-256 digest and signed JWT bearer tokens; validates `Authorization: Bearer TOKEN` before each WebSocket upgrade, rejecting missing or invalid credentials with HTTP 401.

Adds opt-in OTLP logging for final agent responses.

Exposes tool dispatch and timing observations to extensions.

Adds portable project trust lookup APIs.

Adds bounded buffering for global operation metrics.

+10 more

rust-v0.158.0-alpha.3 NOTES ↗ CODE ↗ PRE-RELEASE

Codex CLI TUI now explains Mermaid rendering fallbacks and records sandbox backend in analytics.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch rust-v0.158.0-alpha.3
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.158.0-alpha.3
```

Shows a dimmed notice in the TUI above preserved Mermaid source when a diagram uses unsupported syntax, exceeds renderer size limits, or does not fit the terminal width — instead of silently falling back.

Records the observed sandbox backend as a `sandbox_backend` metric tag in command execution analytics, distinguishing unknown/missing backends from explicitly unsandboxed processes.

WAS THIS USEFUL? 👍 👎

Graphify Labs Graphify

Feedback ⓘ v0.9.67 NOTES ↗

RANK

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

> Graphify expanded language coverage with five new extractors and PHP closure graphing, while also locking down output determinism for reproducible knowledge-graph builds.

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

01 **PHP closure extraction with semantic route naming** NEW

70

Extracts PHP closures — anonymous `function(){…}`, arrow `fn()=>…`, and closure arguments — as graph nodes with inner call capture in all positions including file scope. Route-definition closures receive a semantic `VERB /path` name that composes nested group() prefixes, while other closures get a stable per-scope ordinal.

— Detailed mechanism and naming scheme but automatic, no user-facing command v0.9.67

v0.9.67 NOTES ↗ LATEST STABLE

Graphify v0.9.67 adds PHP closure extraction with semantic route naming for anonymous functions, arrow functions, and closure arguments.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.9.67 https://github.com/Graphify-Labs/graphify.git
# already have the repo? check out this version:
$ git checkout v0.9.67
```

Extracts PHP closures — anonymous `function(){…}`, arrow `fn()=>…`, and closure arguments — as graph nodes with inner call capture in all positions including file scope; route-definition closures receive a semantic `VERB /path` name (composing nested group() prefixes), and other closures get a stable per-scope ordinal.

WAS THIS USEFUL? 👍 👎

Anthropic Claude Code

Feedback ⓘ v2.1.281 NOTES ↗

CODE ↗ RANK

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

> Claude Code's v2.1.281 release strengthens the Claude apps gateway with Bedrock IAM role assumption and guardrails, extends `claude plugin validate` to catch more misconfigurations, and adds MCP URL-mode elicitation alongside several smaller config and UI additions.

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

01 **Bedrock upstream access controls in gateway** NEW 80

Adds `assume_role` on Claude apps gateway Bedrock upstreams so the gateway calls Bedrock as an IAM role assumed through STS, supporting cross-account and per-developer sessions. Adds `guardrail: {id, version}` on Bedrock upstreams to apply an Amazon Bedrock guardrail to every request routed through them.

— Names both config keys and the STS/guardrail mechanism v2.1.281

02 **Expanded plugin validation checks** IMPROVED 65

Extends `claude plugin validate` to report `.mcp.json` entries that would be silently dropped at load, undeclared `${user_config.*}` references, and insecure URLs.

— Names command and three specific new checks v2.1.281

03 **Attribution suppression setting** NEW 60

Adds `"attribution": false` in `settings.json` to suppress all commit and PR attribution in Claude Code output.

Suppress commit and PR attribution in a shared settings file so Claude Code output never includes git blame or PR references

```json
{
  "attribution": false
}
```

— Exact config key, file, and example provided v2.1.281

THINNER COVERAGE BELOW

04 **Gateway config additions for policy and telemetry** NEW

55

Adds `blockReadsOutsideWorkingDirectories` and `disableBypassPermissionsMode` to `desktop` policy blocks in the Claude apps gateway config for newer Claude Desktop keys, and adds `telemetry.resource_attributes` to attach fixed labels to telemetry from Claude Desktop and `/login` sessions.

— Names three config keys but describes each thinly v2.1.281

05 **MCP URL-mode elicitation** NEW 50

Adds MCP URL-mode elicitation on 2026-07-28 protocol connections, letting MCP servers trigger a browser-based flow without leaving a waiting dialog on screen.

— Explains mechanism but no direct usage step v2.1.281

06 **Auto mode recommendation in /insights** NEW 50

Adds an auto mode recommendation to `/insights` that estimates how many recent permission prompts auto mode could have handled.

— Names the command and what it estimates v2.1.281

07 **Scrollbar in installed lists** IMPROVED 35

Adds a scrollbar to the `/skills`, `/mcp`, and `/plugin` Installed lists in fullscreen mode.

— Minor UI tweak with limited detail v2.1.281

v2.1.281 NOTES ↗ CODE ↗ LATEST STABLE

Claude Code v2.1.281 adds gateway IAM role assumption, Bedrock guardrails, MCP elicitation, and richer plugin validation

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.1.281 https://github.com/anthropics/claude-code.git
# already have the repo? check out this version:
$ git checkout v2.1.281
```

└──▷ USE IT

Suppress commit and PR attribution in a shared settings file so Claude Code output never includes git blame or PR references

settings.json

```
{
  "attribution": false
}
```

Adds `blockReadsOutsideWorkingDirectories` and `disableBypassPermissionsMode` to `desktop` policy blocks in the Claude apps gateway config for newer Claude Desktop keys

Adds `assume_role` on Claude apps gateway Bedrock upstreams so the gateway calls Bedrock as an IAM role assumed through STS, supporting cross-account and per-developer sessions

Adds `guardrail: {id, version}` on Claude apps gateway Bedrock upstreams to apply an Amazon Bedrock guardrail to every request routed through them

Adds `telemetry.resource_attributes` to the Claude apps gateway config to attach fixed labels to telemetry from Claude Desktop and `/login` sessions

Adds `"attribution": false` in `settings.json` to suppress all commit and PR attribution in Claude Code output

+4 more

WAS THIS USEFUL? 👍 👎

The Open Engine Zeroshot

Feedback ⓘ v10.8.0 NOTES ↗

RANK

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

> Zeroshot v10.8.0 lets a local UI read run history from any configured target, sharing a common validation contract across direct, hosted and Cloud backends.

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

01 **Local UI reads run history from any target** NEW 45

A local Zeroshot UI can now read canonical run history from a configured direct or hosted target, with a shared validation and response-bounds contract applied consistently across targets, Cloud, and the UI BFF.

— Describes behaviour and scope but no config keys or commands given v10.8.0

v10.8.0 NOTES ↗ LATEST STABLE

Zeroshot v10.8.0 lets a local UI read run history from any configured direct or hosted target.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v10.8.0 https://github.com/the-open-engine/zeroshot.git
# already have the repo? check out this version:
$ git checkout v10.8.0
```

WAS THIS USEFUL? 👍 👎

All Hands AI OpenHands

Feedback ⓘ v1.23.0 NOTES ↗

RANK

OpenHands is an open-source software-development agent that writes code, runs commands, and browses the web.

> OpenHands added script bundle automations to Agent Canvas with dedicated log viewing, alongside a universal macOS DMG installer and two new UI themes.

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

01 Script bundle automations in Agent Canvas NEW

77

Automations can now run a script bundle that executes an entrypoint directly instead of starting a conversation; run records show the message `No conversation — this run executed a script. Use View logs for its output.` A new View logs action on automation run records reads the script's output, resolved from the run's sandbox on cloud backends. The automation detail view gains a Script section, replacing the prompt section for prompt-less automations, which lists the bundle's files.

Read the output of a script automation run before its sandbox is deleted on a cloud backend.

> 📍 In Agent Canvas, open the automation run record and click View logs while the run is still in progress or immediately after it finishes to capture the script output before the sandbox is cleaned up.

Inspect which files and entrypoint a script bundle automation will execute.

> 📍 In Agent Canvas, open the automation's detail view and read the Script section to see the bundle's files and entrypoint.

— Names UI elements and exact run message, with usage steps product docs

THINNER COVERAGE BELOW

02 Universal macOS DMG installer NEW

38

OpenHands ships a universal macOS DMG installer for the desktop app with per-architecture bundled runtimes.

— Names the installer and per-arch detail but no usage steps v1.23.0

03 Light+ and Solarized Light themes NEW

25

The UI now offers Light+ and Solarized Light theme options.

— Bare mention of two theme names, no further detail v1.23.0

└──▷ ALSO FROM THESE RELEASES

v1.23.0 NOTES ↗ LATEST STABLE

OpenHands v1.23.0 adds a universal macOS DMG with per-arch bundled runtimes and Light+/Solarized Light theme support.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.23.0
https://github.com/OpenHands/OpenHands.git
# already have the repo? check out this version:
$ git checkout v1.23.0
```

Adds a universal macOS DMG installer with per-architecture bundled runtimes for the desktop app.

Adds Light+ and Solarized Light theme options to the UI.

WAS THIS USEFUL? 👍 👎

Zed

Feedback ⓘ v1.21.0 NOTES ↗ RANK

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

> Zed v1.21.0 adds a language server command picker with `showDocument` support, a new `agent.prevent_idle_sleep` setting, BYOK access to Claude Opus 5.5 and the GPT-6 family, SuperGrok sign-in for Grok models, and rendering performance improvements across the editor and Markdown surfaces.

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

01 **Comment empty lines toggle option** NEW 75

Adds a `comment_empty_lines` parameter to the `editor::ToggleComments` keybinding action — set `true` to comment blank lines or `false` to skip them; Zed's default keymap uses `true` while the VS Code keymap uses `false`.

Control whether blank lines are commented when toggling comments over a multiline selection — handy when your team style guide requires skipping empty lines.

json

```
// In your Zed keymap.json:
{
  "context": "Editor",
  "bindings": {
    "ctrl-/": ["editor::ToggleComments", {"comment_empty_lines":
false}]
  }
}
```

— Exact keybinding parameter with default behavior and runnable example v1.21.0

02 **Read-only file pattern inheritance** NEW 75

Adds support for `"..."` in `read_only_files` project settings so inherited read-only patterns can be extended rather than fully repeated.

Extend a shared project's inherited read-only file patterns without having to repeat them — useful for enforcing file-protection policies across nested workspaces.

```json
// In your project-level .zed/settings.json:
{
  "read_only_files": ["...", "generated/**", "vendor/**"]
}
```

— Exact config key and syntax with a runnable example v1.21.0

03 **Prevent idle sleep during agent threads** NEW 70

Adds the `agent.prevent_idle_sleep` setting, enabled by default, to stop the system from sleeping while agent threads are running — useful for overnight scans or long code-generation tasks.

Prevent your workstation from sleeping during long-running agent tasks — useful when running overnight security scans or extended code-generation threads.

```json
// In your Zed settings.json:
{
  "agent": {
    "prevent_idle_sleep": true
  }
}
```

— Names exact config key, default, and gives runnable config example v1.21.0

04 **Language server tooling improvements** NEW 65

Adds a language server command picker, support for language servers to open files and URLs via `showDocument` requests, and the ability to run language-server actions directly from actionable inlay hints.

— Names three surfaces but no exact commands to invoke them v1.21.0

05 **New AI model support and BYOK providers** NEW 65

Adds BYOK support for Claude Opus 5.5 with an Anthropic API key, BYOK support for GPT-6 Sol, GPT-6 Luna, and GPT-6 Astra with an OpenAI API key, support for DeepSeek Flash 4.1, and Z.ai GLM added to the Mistral provider.

— Names every model and required API key but no setup steps v1.21.0

06 **Project Panel context menu shortcuts** NEW 65

Adds `menu` (Linux and Windows) and `shift-f10` (all platforms) shortcuts to open the context menu for the selected Project Panel entry.

— Exact keybindings and platform scope named v1.21.0

THINNER COVERAGE BELOW

07 **Rename active Terminal Thread action** NEW 55

Adds the `agent: rename selected thread` action for renaming the active Terminal Thread from the Agent Panel.

— Names the action but not its keybinding or exact steps v1.21.0

08 **CodeLens foreground theme key** NEW 50

Adds the `editor.code_lens.foreground` theme key so CodeLens text color can be customized independently of other editor colors.

— Names the theme key but gives no example usage v1.21.0

09 **Rendering performance improvements** IMPROVED 45

Improves editor rendering performance in syntax-highlighted files, especially with the minimap enabled, and improves rendering performance for Markdown code blocks in the Agent Panel, hover popovers, and Markdown preview.

— Names affected surfaces but no metrics or benchmarks v1.21.0

10 **SuperGrok sign-in for Grok models** NEW 40

Adds SuperGrok sign-in so SuperGrok subscribers can use Grok models in the Agent Panel.

— States the capability without sign-in mechanics v1.21.0

11 **Emmet extension install prompt** NEW 40

Adds a prompt to install the Emmet extension when opening files in Emmet-supported languages.

— Describes trigger but no configuration detail v1.21.0

12 **Threads Sidebar width setting** NEW 35

Adds the `agent.threads_sidebar_default_width` setting to configure the width of the Threads Sidebar.

— Bare setting name with no default value or example v1.21.0

13 **Commit message editor context menu** NEW 35

Adds Cut , Copy , and Paste context menu actions to the commit message editor.

— Names the actions but nothing beyond that v1.21.0

v1.21.0 NOTES ↗ LATEST STABLE

Zed v1.21.0 adds a language server command picker, `agent.prevent_idle_sleep`, BYOK for Claude Opus 5.5 and GPT-6 models, and SuperGrok sign-in.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.21.0 https://github.com/zed-
industries/zed.git
# already have the repo? check out this version:
$ git checkout v1.21.0
```

└──▷ USE IT

Prevent your workstation from sleeping during long-running agent tasks — useful when running overnight security scans or extended code-generation threads.

~/.config/zed/settings.json

```
// In your Zed settings.json:
{
  "agent": {
    "prevent_idle_sleep": true
  }
}
```

Control whether blank lines are commented when toggling comments over a multiline selection — handy when your team style guide requires skipping empty lines.

~/.config/zed/keymap.json

```
// In your Zed keymap.json:
{
  "context": "Editor",
  "bindings": {
    "ctrl-/": ["editor::ToggleComments", {"comment_empty_lines":
false}]
  }
}
```

Extend a shared project's inherited read-only file patterns without having to repeat them — useful for enforcing file-protection policies across nested workspaces.

.zed/settings.json

```
// In your project-level .zed/settings.json:
{
  "read_only_files": ["...", "generated/**", "vendor/**"]
}
```

Adds the `agent.prevent_idle_sleep` setting (enabled by default) to prevent the system from sleeping while agent threads are running.

Adds the `agent.threads_sidebar_default_width` setting to configure the width of the Threads Sidebar.

Adds a `comment_empty_lines` parameter to the `editor::ToggleComments` keybinding action — set `true` to comment blank lines or `false` to skip them; Zed's default keymap uses `true`, VS Code keymap uses `false`.

Adds `editor.code_lens.foreground` theme key for customizing CodeLens text color independently.

Adds support for `"..."` in `read_only_files` project settings so inherited read-only patterns can be extended.

+13 more

WAS THIS USEFUL? 👍 👎

Block Goose

Feedback ⓘ v1.52.0 NOTES ↗ CODE ↗ RANK

Goose is an open-source AI agent with a desktop app and CLI for coding and other automated tasks.

> Goose v1.52.0 adds live voice conversations to the desktop app, enforces new recipe parameter limits, and expands its provider and model lineup with Z.AI, Decisions, and several new models.

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

01 **Recipe parameter limits enforcement** BREAKING 75

Goose now enforces recipe parameter limits: max 32 params, 200 select options, and 128 KiB per recipe.

– Exact numeric limits given but no migration guidance v1.52.0

02 **New AI providers and model support** NEW 65

Adds a Decisions provider crate with `OpenRouter` and Jev implementations, a `Z.AI Coding Plan` provider with streaming tool calls, and support for a custom base URL for the OpenAI provider in the SDK. Also adds support for `Opus 5.5`, GPT-6-sol, and GPT-6-luna models.

– Names providers and models but no config steps v1.52.0

THINNER COVERAGE BELOW

03 **Live voice conversations in desktop app** NEW 30

Adds live voice conversations in the desktop app.

– Bare description, no mechanism or UI path v1.52.0

04 **Full-screen settings view in desktop app** IMPROVED 30

Redesigns settings as a full-screen view in the desktop app.

– UI change noted without further detail v1.52.0

v1.52.0 NOTES ↗ CODE ↗ LATEST STABLE

Goose v1.52.0 adds live voice conversations, Z.AI and Decisions providers, and recipe parameter limits enforcement.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v1.52.0 https://github.com/aaif-goose/goose.git
# already have the repo? check out this version:
$ git checkout v1.52.0
```

Enforces recipe parameter limits of max 32 params, 200 select options, and 128 KiB per recipe.

Adds live voice conversations in the desktop app.

Adds a Decisions provider crate with OpenRouter and Jev implementations.

Adds Z.AI Coding Plan provider with streaming tool calls.

Supports custom base URL for the OpenAI provider in the SDK.

+2 more

WAS THIS USEFUL? 👍 👎

Amazon Kiro

Feedback ⓘ

CLI: Reasoning Controls and Cloud Sessions: Configure thinking and effort from /model an

BLOG ↗ RANK

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

> Kiro CLI 2.23 reworks reasoning controls around `/model`, lets V3 cloud sessions start before a repository is attached, and adds a batch of session-management and performance improvements across the CLI.

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

01 Reasoning controls consolidated into /model

BREAKING

75

Adds reasoning controls (thinking and effort) directly inside `/model`, replacing `/effort` as the primary interface for those settings. In V2, bare `/effort` is now a legacy command that shows a deprecation warning and opens the Effort setting in `/model` instead of acting directly, though `/effort <level>` remains supported.

— Names the changed commands but no full example of new flow

CLI: Reasoning Controls and Cloud Sessi…

02 Animated title-bar spinner via chat.terminalTitle

NEW

70

Adds the `chat.terminalTitle` config key to enable an animated title-bar spinner reflecting agent activity, so you can glance at the terminal to see whether Kiro is actively working.

Enable the animated title-bar spinner so you can glance at your terminal and see at a glance whether Kiro is actively working.

```yaml
chat.terminalTitle: true
```

— Names exact config key and shows the value to set it

CLI: Reasoning Controls and Cloud Sessi…

03 Session management improvements in CLI

IMPROVED

70

Adds a `/sessions` filter in 2.23.1 to show only main sessions, hiding Tangent, rewind, subagent, and other derived child sessions. Adds `--resume-id` error reporting in

2.23.1 so classic mode reports an error for missing, unreadable, empty, or V2-only sessions instead of silently starting a new conversation.

— Names both surfaces with clear before/after behavior

CLI: Reasoning Controls and Cloud Sessi…

04 **V3 cloud sessions before repository connection** NEW

65

V3 cloud sessions can now start before connecting a repository, with the connection screen marking the repository as optional. The new `/repo` command attaches a repository to an already-running V3 cloud session.

Start a V3 cloud session immediately and attach a repository later once you know which one to use.

```
$ /repo
```

— Names command and includes usage example CLI: Reasoning Controls and Cloud Sessi…

THINNER COVERAGE BELOW

05 **Performance and terminal experience improvements** IMPROVED

55

Improves streaming response rendering with smoother output and lower CPU usage during long conversations, improves fullscreen wheel scrolling efficiency, adds ConEmu support so progress indicators work when ANSI processing is enabled, reduces Kiro CLI installation size via compressed bundled runtimes, and improves `/sessions` open speed in V3 especially with large session histories.

— Concrete list of improvements but mostly under-the-hood, no action needed

CLI: Reasoning Controls and Cloud Sessi…

06 **Denied tool feedback in V3** NEW

45

When a tool call is denied in V3, the text entered while denying it is now sent to the model so it can adjust its next action.

— Explains mechanism but no config or command to try it

CLI: Reasoning Controls and Cloud Sessi…

07 **Response attribution in V3** NEW

30

Server-provided content is now kept separate from the model's own responses in V3.

— Bare description with no mechanism detail or surface

CLI: Reasoning Controls and Cloud Sessi…

└──▷ BREAKING ON UPGRADE

! In V2, bare `/effort` is now a legacy command: it shows a deprecation warning and opens the Effort setting in `/model` instead of acting directly. `/effort <level>` remains supported.

`CLI: Reasoning Controls and Cloud Sessions: Configure thinking and effort from /model`

BLOG ↗ LATEST STABLE

Kiro CLI 2.23 adds reasoning controls in `/model`, cloud sessions before repo connect, denied tool feedback, and response attribution.

└──▷ **TRY IT**

Start a V3 cloud session immediately and attach a repository later once you know which one to use.

```
$ /repo
```

Enable the animated title-bar spinner so you can glance at your terminal and see at a glance whether Kiro is actively working.

```yaml
chat.terminalTitle: true
```

Adds `chat.terminalTitle` config key to enable an animated title-bar spinner reflecting agent activity.

Adds `/repo` command to attach a repository to an already-running V3 cloud session.

Enables V3 cloud sessions to start before connecting a repository; the connection screen marks the repository as optional.

Adds reasoning controls (thinking and effort) directly inside `/model`, replacing `/effort` as the primary interface for those settings.

Adds denied tool feedback in V3: text entered when denying a tool call is sent to the model so it can adjust its next action.

+8 more

└──▷ **BREAKING ON UPGRADE**

! In V2, bare `/effort` is now a legacy command: it shows a deprecation warning and opens the Effort setting in `/model` instead of acting directly. `/effort <level>` remains supported.

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

Mastra

Feedback ⓘ @mastra/core@1.69.0 NOTES ↗ RANK

Mastra is a TypeScript framework for AI applications and agents, providing workflows, agent memory, RAG, tool calling, and evals.

> Mastra 1.69.0 introduces a new Classifier primitive for AI-based evaluation with a safety-gating processor and typed workflow steps, adds ten generated SaaS tool providers via @mastra/connect, and lets tools hand off long-running work to Mastra for background tracking; docs also detail platform environment variable access for deployed services.

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

01 **ClassifierProcessor safety gate for agents** NEW 90

Adds `ClassifierProcessor` at `@mastra/core/processors` to enforce typed classifier policies on agent `inputProcessors`, output, and streaming content, with an explicit abort() callback. It fails closed by default, aborting requests if the classifier call fails; pass `errorStrategy: 'warn'` to restore the previous fail-open behavior.

Gate agent input with a safety classifier that aborts requests scoring above an unsafe probability threshold.

```typescript
import { Agent } from '@mastra/core/agent';
import { Classifier } from '@mastra/core/classifier';
import { ClassifierProcessor } from '@mastra/core/processors';

const safety = new Classifier({
  id: 'safety',
  model,
  questions: {
    unsafe: { type: 'boolean', criteria: { true: 'Unsafe', false:
'Safe' } },
  },
});

const agent = new Agent({
  id: 'support-agent',
  name: 'Support agent',
  instructions: 'Answer support questions.',
  model: 'openai/gpt-5-mini',
  inputProcessors: [
    new ClassifierProcessor({
      classifier: safety,
      onResult: (answers, { abort }) => {
        if (answers.unsafe.probability > 0.8) abort('Rejected by
safety policy');
      },
    }),
  ],
});
```

— Runnable safety-gate example with exact override flag `@mastra/core@1.69.0`

02 **Classifier-driven typed workflow steps** NEW 81

Adds registered classifiers as typed workflow steps via .classifier() with fluent and dynamic graph support, exposing typed answers and token usage for use in .branch() and conditional control flow.

Route workflow branching with a typed classifier step so billing and support paths are selected by AI evaluation rather than hand-coded rules.

```typescript
import { Classifier } from '@mastra/core/classifier';

const router = new Classifier({
  id: 'router',
  model,
  questions: {
    route: {
      type: 'choice',
      criteria: { billing: 'Billing request', support: 'Support request' },
    },
  },
});

workflow
  .map({ message: { initData: true, path: 'message' } })
  .classifier(router)
  .branch([
    [async ({ inputData }) => inputData.answers.route.choice === 'billing', billingStep],
    [async ({ inputData }) => inputData.answers.route.choice === 'support', supportStep],
  ]);
```

— Working branch-routing example demonstrates the mechanism `@mastra/core@1.69.0`

03 **Background tool completion tracking** NEW 80

Adds context.background.adopt({ completion, cancel }) in tool execute() so tools can return an acknowledgement immediately while handing a long-running native background operation to Mastra for completion and cancellation tracking.

Return an immediate acknowledgement from a long-running research tool while Mastra tracks the background operation to completion.

```typescript
execute: async (input, context) => {
  const operation = startResearch(input, context.abortSignal);
  if (context.background) {
    context.background.adopt({
      completion: operation.finished,
      cancel: reason => operation.cancel(reason),
    });
    return { answer: 'Research started' };
  }
  return await operation.finished;
},
```

— Concrete API with runnable example of the handoff `@mastra/core@1.69.0`

04 **Classifier primitive for fixed-option AI evaluation** NEW

79

Adds Classifier primitive at `@mastra/core/classifier` for fixed-option AI evaluation via classifier.evaluate(), supporting `choice` and `boolean` question types with per-answer probability scoring. Classifiers register via new Mastra({ classifiers }) and expose a management API (`getClassifier` , `getClassifierById` , `listClassifiers` , `addClassifier` , `removeClassifier`); evaluations without an active trace start a root `CLASSIFIER_EVALUATION` span.

— Names full management API but lacks a direct runnable example `@mastra/core@1.69.0`

05 **Playground UI layout components restructured** BREAKING

73

`@mastra/playground-ui/lib/springs` has been removed and `FluidHoverHighlight` now accepts only `hover` and `className` props; `PageLayoutRoot` , `MainContentLayout` , `MainContentContent` , and multiple `AppShell` header-related props/contexts have been removed — migrate to the new `PageLayout` component with `breadcrumbs` , `headerActions` , and `actionRow` props.

— Names removed and replacement props but no migration snippet `@mastra/core@1.69.0`

06 **Ten SaaS tool providers in @mastra/connect** NEW

70

Adds ten generated SaaS tool providers in `@mastra/connect@0.3.0` — Slack, GitHub, Google Mail, Google Calendar, Fireflies, PostHog, Stripe, Discord, Twitter/X, and HubSpot — built via Nango templates; connect() resolves the right tools per request with no extra configuration.

— Lists all ten providers but no code sample `@mastra/core@1.69.0`

07 **Custom root span names for workflow and agent runs** NEW

65

Adds `rootSpanName` to `tracingOptions` on workflow and agent runs so each run can set its own root span name instead of the default `workflow run: '<id>'` / `agent run: '<id>'`; the generated client request types also expose `rootSpanName` so it can be sent via run.startAsync().

— Named config key but no usage example given `@mastra/core@1.69.0`

---

08 **Trace query grouping deprecated in favor of queryThreads**

DEPRECATED

65

The `group` option on `TraceQueryRequest` / queryTraces() is deprecated, remaining functional until the next major release; migrate to `QueryThreadsInput` / queryTraceThreads() / `queryThreads` for grouped trace results.

— Clear migration target named, no code example `@mastra/core@1.69.0`

---

09 **Retries for Inngest workflows and durable agents** IMPROVED

60

Adds a `retries` option to Inngest workflows and durable agents so runs survive process restarts and redeploys by retrying and continuing past already-completed steps.

— Names the option but no config example shown `@mastra/core@1.69.0`

└──▷ **BREAKING ON UPGRADE**

! `@mastra/playground-ui`: `@mastra/playground-ui/lib/springs` has been removed and `FluidHoverHighlight` now accepts only `hover` and `className` props.

! `@mastra/playground-ui`: `PageLayoutRoot`, `MainContentLayout`, `MainContentContent`, and multiple `AppShell` header-related props/contexts have been removed; migrate to the new `PageLayout` with `breadcrumbs`, `headerActions`, and `actionRow`.

! The `group` option on `TraceQueryRequest` / queryTraces() is deprecated (still functional until the next major release); migrate to `QueryThreadsInput` / queryTraceThreads() / `queryThreads` for grouped trace results.

! `ClassifierProcessor` now fails closed by default: when the classifier call fails the request is aborted. Pass `errorStrategy: 'warn'` to restore the previous fail-open behavior.

`@mastra/core@1.69.0` NOTES ↗ LATEST STABLE

Mastra 1.69.0 adds Classifier primitives, `ClassifierProcessor` safety gates, background tool adoption, and ten new `@mastra/connect` SaaS providers.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch @mastra/core@1.69.0 https://github.com/mastra-ai/mastra.git
# already have the repo? check out this version:
$ git checkout @mastra/core@1.69.0
```

└──▷ **USE IT**

Gate agent input with a safety classifier that aborts requests scoring above an unsafe probability threshold.

```typescript
import { Agent } from '@mastra/core/agent';
import { Classifier } from '@mastra/core/classifier';
import { ClassifierProcessor } from '@mastra/core/processors';

const safety = new Classifier({
  id: 'safety',
  model,
  questions: {
    unsafe: { type: 'boolean', criteria: { true: 'Unsafe', false: 'Safe' } },
  },
});

const agent = new Agent({
  id: 'support-agent',
  name: 'Support agent',
  instructions: 'Answer support questions.',
  model: 'openai/gpt-5-mini',
  inputProcessors: [
    new ClassifierProcessor({
      classifier: safety,
      onResult: (answers, { abort }) => {
        if (answers.unsafe.probability > 0.8) abort('Rejected by safety policy');
      },
    }),
  ],
});
```

Route workflow branching with a typed classifier step so billing and support paths are selected by AI evaluation rather than hand-coded rules.

```typescript
import { Classifier } from '@mastra/core/classifier';

const router = new Classifier({
  id: 'router',
  model,
  questions: {
    route: {
      type: 'choice',
      criteria: { billing: 'Billing request', support: 'Support
request' },
    },
  },
});

workflow
  .map({ message: { initData: true, path: 'message' } })
  .classifier(router)
  .branch([
    [async ({ inputData }) => inputData.answers.route.choice ===
'billing', billingStep],
    [async ({ inputData }) => inputData.answers.route.choice ===
'support', supportStep],
  ]);
```

Return an immediate acknowledgement from a long-running research tool while Mastra tracks the background operation to completion.

```typescript
execute: async (input, context) => {
  const operation = startResearch(input, context.abortSignal);
  if (context.background) {
    context.background.adopt({
      completion: operation.finished,
      cancel: reason => operation.cancel(reason),
    });
    return { answer: 'Research started' };
  }
  return await operation.finished;
},
```

Adds `Classifier` primitive at `@mastra/core/classifier` for fixed-option AI evaluation with classifier.evaluate(), supporting `choice` and `boolean` question types with per-answer probability scoring.

Adds classifier registration on new Mastra({ classifiers }) with management API: `getClassifier`, `getClassifierById`, `listClassifiers`, `addClassifier`, `removeClassifier`; evaluated classifiers without an active trace start a root `CLASSIFIER_EVALUATION` span.

Adds `ClassifierProcessor` at `@mastra/core/processors` to enforce typed classifier policies on agent `inputProcessors`, output, and streaming content, with explicit abort() callback and fail-closed default; pass `errorStrategy: 'warn'` to restore fail-open behavior.

Adds registered classifiers as typed workflow steps via .classifier() with fluent and dynamic graph support, exposing typed answers and token usage for use in .branch() and conditional control flow.

Adds context.background.adopt({ completion, cancel }) in tool execute() so tools can return an acknowledgement immediately while handing a long-running native background operation to Mastra for completion/cancellation tracking.

+5 more

└──▷ **BREAKING ON UPGRADE**

- ! `@mastra/playground-ui`: `@mastra/playground-ui/lib/springs` has been removed and `FluidHoverHighlight` now accepts only `hover` and `className` props.
- ! `@mastra/playground-ui`: `PageLayoutRoot`, `MainContentLayout`, `MainContentContent`, and multiple `AppShell` header-related props/contexts have been removed; migrate to the new `PageLayout` with `breadcrumbs`, `headerActions`, and `actionRow`.
- ! The `group` option on `TraceQueryRequest` / queryTraces() is deprecated (still functional until the next major release); migrate to `QueryThreadsInput` / queryTraceThreads() / `queryThreads` for grouped trace results.
- ! `ClassifierProcessor` now fails closed by default: when the classifier call fails the request is aborted. Pass `errorStrategy: 'warn'` to restore the previous fail-open behavior.

WAS THIS USEFUL? 👍 👎

PydanticAI

Feedback ⓘ v2.49.0 NOTES ↗ RANK

PydanticAI is a framework that builds type-safe Python agents with dependency injection, model integrations, tools, and structured outputs.

> PydanticAI v2.49.0 adds GitHub Copilot OAuth support, explicit boolean semantics for structured outputs, a blocking realtime reply helper, and two refinements to TypeSafeModel's field handling.

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

01 BoolCriteria for explicit bool field semantics NEW

80

Adds a `BoolCriteria` class to explicitly declare what `True` and `False` mean for a `bool` field, with equivalent support via a `True` / `False` `Enum`. For example, annotating a field with BoolCriteria(yes='All OWASP Top 10 items were addressed', no='One or more OWASP Top 10 items remain open') gives an LLM-backed evaluator explicit criteria for each outcome.

Annotate a bool field with human-readable yes/no semantics so an LLM-backed evaluator knows exactly what True and False represent.

python

```python
from pydantic_ai import BoolCriteria
from pydantic import BaseModel
from typing import Annotated

class Review(BaseModel):
    passed_security_audit: Annotated[
        bool,
        BoolCriteria(
            yes='All OWASP Top 10 items were addressed',
            no='One or more OWASP Top 10 items remain open',
        ),
    ]
```

— Concrete class with runnable annotation example v2.49.0

02 RealtimeSession.wait_for_reply() NEW

70

Adds RealtimeSession.wait_for_reply() to block until the realtime session produces a reply, allowing code like reply = await session.wait_for_reply() instead of manually consuming events from the session.

Await a complete realtime reply in an async flow rather than manually consuming events from the session.

```python
reply = await session.wait_for_reply()
```

— Named method with a runnable code snippet v2.49.0

THINNER COVERAGE BELOW

03 **GitHub Copilot OAuth device flow** NEW 48

Adds `GitHubCopilotOAuthFlow` to support the device authorization flow when authenticating with GitHub Copilot models.

— Names the class but no usage example given v2.49.0

04 **TypeSafeModel field handling refinements** IMPROVED 43

`TypeSafeModel` now uses the developer's own `None` field description as the `none of these` option label, and offers whole-number fields that have no rubric as a `Choice`.

— Named surface but no example or mechanism detail v2.49.0

v2.49.0 NOTES ↗ LATEST STABLE

PydanticAI v2.49.0 adds GitHub Copilot OAuth, `BoolCriteria`, RealtimeSession.wait_for_reply(), and richer `TypeSafeModel` semantics.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.49.0 https://github.com/pydantic/pydantic-ai.git
# already have the repo? check out this version:
$ git checkout v2.49.0
```

└──▷ USE IT

Await a complete realtime reply in an async flow rather than manually consuming events from the session.

```python
reply = await session.wait_for_reply()
```

Annotate a bool field with human-readable yes/no semantics so an LLM-backed evaluator knows exactly what True and False represent.

```python
from pydantic_ai import BoolCriteria
from pydantic import BaseModel
from typing import Annotated

class Review(BaseModel):
    passed_security_audit: Annotated[
        bool,
        BoolCriteria(
            yes='All OWASP Top 10 items were addressed',
            no='One or more OWASP Top 10 items remain open',
        ),
    ]
```

Adds `GitHubCopilotOAuthFlow` for device authorization flow when authenticating with GitHub Copilot models.

Adds `BoolCriteria` class to explicitly declare what `True` and `False` mean for a `bool` field, with equivalent support via a `True` / `False` `Enum`.

Adds RealtimeSession.wait_for_reply() method to block until the realtime session produces a reply.

Extends `TypeSafeModel` to use the developer's own `None` field description as the `none of these` option label.

Extends `TypeSafeModel` to offer whole-number fields that have no rubric as a `Choice`.

WAS THIS USEFUL? 👍 👎

Feynman

Feedback ⓘ v0.5.2 NOTES ↗ RANK

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

> Feynman rebuilt itself on an unmodified Pi runtime this window, swapping its own workbench, ranking, and browser-auth machinery for Pi-native session, telemetry, and subagent primitives, while its literature-search tool gained Semantic Scholar and OpenAlex semantic search alongside a narrower source scope and hard result caps.

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

01 **Opt-out telemetry with status visibility** NEW 90

Introduces anonymous research telemetry emitted by the Pi extension and CLI: events include `feynman_session_started`, `feynman_workflow_started`, `feynman_workflow_completed`, `feynman_tool_used`, `feynman_command_started`, `feynman_command_completed`, `feynman_command_failed`, and PostHog `$ai_generation` (model, provider, token counts, latency, HTTP status, stop reason); prompts and content are never sent, and traces no longer carry command line, host name, or user name. `feynman status` shows whether telemetry is on; set `DO_NOT_TRACK=1` or `FEYNMAN_TELEMETRY=off` to disable both CLI and Pi runtime events, and a one-time first-run notice on stderr (suppressed in scripts/CI) explains what is collected.

Check telemetry status and disable all event collection in CI pipelines where you do not want research activity reported.

```
$ feynman status
# To disable:
export DO_NOT_TRACK=1
```

Opt out of all telemetry in a CI environment where you don't want usage data sent to PostHog.

```
$ FEYNMAN_TELEMETRY=off feynman deepresearch "mechanistic interpretability"
```

Opt out of all telemetry before running a research query that includes sensitive prompt text.

```
$ FEYNMAN_TELEMETRY=off feynman "what do we know about
adversarial prompt injection"
```

— Names every event and both disable env vars, runnable. v0.5.1 v0.3.49

02 **CLI session management flags** NEW 90

Adds `--resume` / `-r` to pick a previous session to resume, `--session <path|id>` to open a specific session, `--fork <path|id>` to fork an existing session, `--no-session` to run with an in-memory session that isn't persisted, `--export <session.jsonl> [out.html]` to export a session to HTML and exit, and `--continue` / `-c` to continue the most recent session (the interactive default). Also supports piping text directly into Feynman as the prompt when stdin isn't a terminal (e.g. `git diff | feynman --no-session`); these flags are now passed through to Pi rather than rejected.

Pipe a git diff directly into Feynman for a one-shot, non-persisted analysis — useful in CI or pre-commit hooks where you want no session state left behind.

```
$ git diff | feynman --no-session
```

Export a saved session to an HTML file for sharing or archiving an investigation.

```
$ feynman --export session.jsonl report.html
```

Resume a previous research session by ID — now that Pi session flags pass through, you can hand off a saved session directly from the CLI.

```
$ feynman --session abc123 "continue the mechanistic
interpretability review"
```

— Every flag named with runnable command examples. v0.5.0

03 **Semantic Scholar and OpenAlex semantic search sources** NEW 85

Adds `source: "semanticscholar"` to `feynman_science_database_search` with `sort: "relevance"` and `sort: "pub_date"` options, and a `semantic:` prefix on `openalex` queries to invoke OpenAlex embedding search (`search.semantic`) instead of keyword search. Supports `SEMANTIC_SCHOLAR_API_KEY` (sent as `x-api-key`) to use your own Semantic Scholar rate limit instead of the shared anonymous

pool, and `OPENALEX_API_KEY` (sent as the `api_key` query parameter, required for regular OpenAlex use since February 2026); missing keys now surface 401/403/429 errors with instructions for obtaining a free key.

Search for seminal ML papers by citation count using the new Semantic Scholar source, with your own API key to avoid shared rate-limit throttling.

```
$ SEMANTIC_SCHOLAR_API_KEY=your_key feynman --prompt 'Find the 10 most-cited papers on test-time compute scaling'
```

Use OpenAlex embedding search to surface conceptually relevant papers that keyword search misses, by prefixing the query with `semantic:`.

```
$ feynman --prompt 'Search openalex semantic: test-time compute scaling and summarise the top results'
```

— Names exact params, env vars and query syntax; missing quota numbers. v0.4.0

---

04 **Web search config migration to pi-web-access** BREAKING

85

Moves web-search configuration to `~/.feynman/agent/web-search.json` (migrated automatically from the old `~/.feynman/web-search.json` on first launch, read by pi-web-access), backed by a private one-hour fetched-page cache at `~/.feynman/agent/web-search-cache/`. Replaces the Feynman-specific `geminiBrowser`, `allowBrowserAuth`, `browserAuth`, and `chromeProfile` keys with pi-web-access's `allowBrowserCookies` and `browserCookies.profile` keys for Gemini Web cookie access.

Allow browser cookies during web searches and rely on the one-hour page cache to reduce redundant fetches.

```json
{"allowBrowserCookies": true}
```

Enable Gemini Web browser-cookie access using the new pi-web-access keys after migrating away from the removed Feynman-specific keys.

```json
{
  "allowBrowserCookies": true,
  "browserCookies": {
    "profile": "Default"
  }
}
```

— Exact file paths and replacement keys given for migration. v0.5.0

05 **Per-subagent model overrides and exclusion config** BREAKING

85

Sets `subagents.agentOverrides.<name>.model` in `~/.feynman/agent/settings.json` (via pi-subagents' `/subagents`) as the per-subagent model override, replacing the removed `/feynman-model` command — remove the key to inherit the main research model, otherwise set via Pi's `/model`. Sets `subagents.agentExcludeDirs` to `['~/.agents']` to stop agent files there from silently replacing Feynman's bundled agents, and adds subagent runtime config at `~/.feynman/agent/extensions/subagent/config.json` with `missions`, `fleetView`, and `asyncByDefault` defaults.

Pin a specific bundled subagent to a different model while leaving all others to inherit the main research model.

```json
{
  "subagents": {
    "agentOverrides": {
      "my-subagent": {
        "model": "gpt-4o"
      }
    },
    "agentExcludeDirs": ["~/.agents"]
  }
}
```

— Exact config keys and file paths for the override. v0.5.0

06 **Legacy CLI commands, tools, and local state removed** BREAKING

85

Removes `feynman rank` (with `--limit`, `--expand-citations`, `--full-text-top`, `--critique-top`, `--synthesize`, `--synthesis-top`, `--synthesis-`

`model`, `--preference-file`, `--source-fixture`, `--output-dir`, `--json`), `feynman paper` (`--fetch-full-text`), `feynman workbench` (`--port`, `--no-open`, `--no-auth`, `--host`), `feynman serve`, `feynman watch`, and `feynman jobs` — all now exit with status 1. Removes the Pi tools `feynman_connector_tools`, `feynman_connector_call`, `feynman_workbench_context`, `feynman_open_chemistry_sketcher`, and `feynman_model_endpoint_call`, replaced by the `/lit` workflow. Removes the `/commands` and `/capabilities` slash commands (use `/help` and `/tools` instead) and Feynman's own `/thinking` command (use Pi's built-in `/thinking`). Feynman no longer creates `~/.feynman/db/`, `~/.feynman/active-org.json`, or `~/.feynman/orgs/`; existing data in those paths is left in place but no longer managed.

— Every removed command and flag named, with exit status. v0.4.0 v0.3.49

---

07 **Unmodified Pi runtime and faster startup** IMPROVED 80

Runs Pi 0.87.1 unmodified — no patching, renaming, or sealed `runtime-workspace.tgz` archive (previously created, hashed, and verified on launch) — cutting `feynman --help` and `feynman doctor` startup to about one second and letting one-shot prompts exit cleanly instead of hanging. Bundles pi-subagents 0.71, pi-web-access 0.31, pi-btw 0.6, and pi-docparser 4.0 as ordinary npm dependencies loaded as Pi packages from Feynman's install directory, and loads bundled agents, skills, and theme from the Feynman package rather than copying them into `~/.feynman/agent` (old copies are deleted on first launch unless edited).

Run a one-shot research prompt in a CI/headless pipeline — the process now exits cleanly when done instead of hanging.

```
$ feynman --prompt="summarize the top three RLHF alternatives published in 2024" > rlhf-brief.txt
```

— Concrete version numbers and startup timing, no user command. v0.5.0

---

08 **npm package renamed back to @companion-ai/feynman** BREAKING 80

The npm package moves from `@advaitpaliwal/feynman` back to `@companion-ai/feynman`; users of the interim 0.3.48 package must run `npm uninstall -g @advaitpaliwal/feynman` then `npm install -g @companion-ai/feynman`.

— Exact migration commands given, fully runnable. v0.3.49

---

09 **OpenAlex result and payload caps** BREAKING 75

Caps `openalex_search_works`, citations, references, authors, and venues results at the tool's `limit` (default 5, max 20) per call, with `max_records` hard-capped at 50 —

down from calls that previously returned 50–100 records — cutting per-call payload from 100–240 KB.

— Exact numeric before/after but no way to opt out. v0.5.2

10 **alphaXiv session status and paper access restriction** IMPROVED

75

Adds `feynman alpha status` to check alphaXiv login state against the alphaXiv server, and restricts `alpha_ask_paper` to arXiv and alphaXiv papers, directing DOI-only papers to `fetch_content` or Europe PMC full text instead.

Verify your alphaXiv session is still valid after a token expiry, especially in a headless CI or SSH environment.

```
$ feynman alpha status
```

— Runnable status command plus named fallback tools. v0.5.2 v0.4.0

11 **Literature source scope narrowed to core databases** BREAKING

73

Narrows `feynman_science_database_search` to OpenAlex, arXiv ID lookup, PubMed, Europe PMC, bioRxiv/medRxiv, and Crossref, removing biomedical, chemistry, genomics, and other non-literature sources. The `arxiv` free-text search source is removed — only arXiv ID lookup remains (bare IDs, URLs, or `arxiv_get_papers:`); use `semanticscholar` or `openalex` for topic searches. `alpha_ask_paper` is now restricted to arXiv and alphaXiv papers, with DOI-only papers directed to `fetch_content` or Europe PMC full text instead.

— Clear before/after scope but no migration command given. v0.5.2 v0.4.0

12 **Default model picks and Pi-routed updates** IMPROVED

73

Recommends Claude Opus 5.5 and GPT-5.6 Terra as default models via `feynman model` and setup, driven by Pi 0.87.1's catalog; premium tiers remain available through `--model` and `/model`. Routes `feynman update` and `feynman packages install` through Pi's own `pi update --extensions` and `pi install`, with optional packages landing in `~/.feynman/agent/npm`.

— Names exact commands and models but no walkthrough. v0.5.0

13 **Researcher/verifier subagent routing and background execution** IMPROVED

70

Routes the researcher agent by source type — Semantic Scholar for general ML/CS, PubMed and Europe PMC for biomedicine, OpenAlex for citation graphs and semantic search — running 2–4 reworded queries per question, and extends the verifier agent to cross-check each cited paper's DOI or arXiv ID against two indexes and flag title/ID mismatches. Runs the `researcher` and `verifier` subagents in the background by default so they receive Feynman's web and literature tools without requiring manual retries.

— Explains routing and cross-check logic, no direct command. v0.5.2 v0.4.0

---

14 **Async subagent workflow API replaces tasks/chain** BREAKING

68

Updates pi-subagents to 0.65.1 with async workflow scripts using `runs.run` / `runs.all`, replacing the obsolete top-level `tasks` / `chain` parameters and the `/chain` / `/parallel` commands; subagent workflows must now use the `runs.run` / `runs.all` interface.

— Names replacement API but shows no example call. v0.3.49

---

15 **Runtime and toolchain version bumps** IMPROVED

65

Moves bundled paper tools to `@companion-ai/alpha-hub@0.1.4` (current alphaXiv OAuth/MCP API and result parser), updates native bundles to Node 24.20.0 and the research runtime compiler to esbuild 0.28.2, and updates the coordinated Pi runtime to 0.85.1, retaining provider identity checks, research-session continuity, bounded compaction, private state files, and tool/telemetry safeguards.

— Version numbers listed but no user action follows. v0.3.49

THINNER COVERAGE BELOW

16 **Web research batching and citation preservation** IMPROVED

55

Updates web research to pi-web-access 0.28.0, adding bounded batch searches, retrievable response IDs, citation preservation, and session-scoped page-answer models.

— Names the mechanism but gives no usage example. v0.3.49

└──▷ ALSO FROM THESE RELEASES

Opt out of telemetry and supply API keys for literature databases before running a research session.

```
$ export DO_NOT_TRACK=1
export OPENALEX_API_KEY=<your-openalex-key>
export SEMANTIC_SCHOLAR_API_KEY=<your-s2-key>
feynman status
```

└──▷ BREAKING ON UPGRADE

- ! The `geminiBrowser`, `allowBrowserAuth`, `browserAuth`, and `chromeProfile` config keys are no longer read; set `allowBrowserCookies` and `browserCookies.profile` in pi-web-access config instead.
- ! Web search config is now read from `~/.feynman/agent/web-search.json`; `~/.feynman/web-search.json` is moved on first launch but any tooling writing to the old path will no longer take effect.
- ! The `/feynman-model` command is removed; use Pi's `/model` for the main model and `subagents.agentOverrides.<name>.model` for per-subagent model selection.
- ! The `feynman workbench` command and its `--port`, `--no-open`, `--no-auth`, `--host` flags are removed along with the local web app and workbench-only tools (`feynman_connector_tools`, `feynman_connector_call`, `feynman_workbench_context`, `feynman_open_chemistry_sketcher`).
- ! The `feynman rank` command (and all its flags: `--limit`, `--expand-citations`, `--full-text-top`, `--critique-top`, `--synthesize`, `--synthesis-top`, `--synthesis-model`, `--preference-file`, `--source-fixture`, `--output-dir`, `--json`) and `feynman paper --fetch-full-text` are removed; removed commands exit with status 1.
- ! The `feynman watch` and `feynman jobs` commands are removed; they exit with status 1 on invocation.
- ! The sealed `.feynman/runtime-workspace.tgz` archive, its hashes, and its lock are no longer created or verified.
- ! The `arxiv` free-text search source in `feynman_science_database_search` is removed; only arXiv ID lookup remains. Use `semanticscholar` or `openalex` for topic searches.
- ! OpenAlex exact-search calls that previously returned 50–100 records now return at most the tool's `limit` (default 5, max 20); `max_records` is hard-capped at 50.
- ! Bundled agents, skills, and the theme are no longer copied into `~/.feynman/agent`; copies left by older releases are deleted on first launch unless you edited them.
- ! Feynman no longer creates `~/.feynman/db/` or `~/.feynman/orgs/`.
- ! The `/feynman-model` slash command is removed; use Pi's `/model` for the main model and pi-subagents' `/subagents` or `subagents.agentOverrides.<name>.model` for per-subagent overrides.
- ! The Feynman-specific `geminiBrowser`, `allowBrowserAuth`, `browserAuth`, and `chromeProfile` keys in `web-search.json` are no longer read; set `allowBrowserCookies` and `browserCookies.profile` in `~/.feynman/agent/web-search.json` instead.
- ! The sealed runtime workspace (`~/.feynman/runtime-workspace.tgz`, its hashes, and its lock) is removed and is no longer verified on launch.

- ! Bundled agents, skills, and theme are no longer copied into `~/.feynman/agent`; copies left by older releases are deleted on first launch unless you edited them.
- ! `feynman serve` (and its `--port`, `--no-open`, `--no-auth`, `--host` flags), `feynman watch`, `feynman jobs`, `feynman rank`, and `feynman paper` are removed; they now exit with status 1.
- ! `feynman_science_database_search` no longer accepts `arxiv_search:` or free-text arXiv queries; arXiv is now ID-only (bare IDs, URLs, or `arxiv_get_papers:`).
- ! The Pi tools `feynman_connector_tools`, `feynman_connector_call`, `feynman_workbench_context`, `feynman_open_chemistry_sketcher`, and `feynman_model_endpoint_call` are removed.
- ! Feynman no longer creates `~/.feynman/active-org.json` or `~/.feynman/orgs/`; existing data in those paths is left in place but no longer managed.
- ! The `/commands` and `/capabilities` slash commands are removed; use `/help` and `/tools` instead.
- ! If you customized `~/.feynman/agent/agents/researcher.md` or `verifier.md`, the new researcher routing and verifier cross-check logic is not applied automatically — you must merge the upstream changes by hand.
- ! The obsolete top-level `tasks` and `chain` parameters and the `/chain` and `/parallel` commands are removed; subagent workflows must now use the `runs.run` / `runs.all` async workflow interface.
- ! Feynman's own `/thinking` command is removed; use Pi's built-in `/thinking` instead.
- ! The npm package has moved from `@advaitpaliwal/feynman` back to `@companion-ai/feynman`; users of the interim 0.3.48 package must run `npm uninstall -g @advaitpaliwal/feynman` then `npm install -g @companion-ai/feynman`.

v0.5.2 NOTES ↗ LATEST STABLE

Feynman v0.5.2 caps OpenAlex results at 20 by default and moves researcher/verifier subagents to background execution.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.5.2 https://github.com/Companion-
Inc/feynman.git
# already have the repo? check out this version:
$ git checkout v0.5.2
```

Caps `openalex_search_works`, citations, references, authors, and venues results at the tool's `limit` (default 5, max 20) per call, with an explicit `max_records` hard-capped at 50, slashing per-call payload from 100-240 KB.

Runs `researcher` and `verifier` subagents in the background by default so they receive Feynman's web and literature tools without requiring manual retries.

Restricts `alpha_ask_paper` to arXiv and alphaXiv papers, directing DOI-only papers to `fetch_content` or Europe PMC full text instead.

v0.5.1 NOTES ↗ STABLE

Feynman v0.5.1 adds opt-out telemetry with `FEYNMAN_TELEMETRY=off` and a first-run notice visible via `feynman status`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.5.1 https://github.com/Companion-
Inc/feynman.git
# already have the repo? check out this version:
$ git checkout v0.5.1
```

└──▷ TRY IT

Opt out of all telemetry in a CI environment where you don't want usage data sent to PostHog.

```
$ FEYNMAN_TELEMETRY=off feynman deepresearch "mechanistic
interpretability"
```

Adds `FEYNMAN_TELEMETRY=off` environment variable (also honours `DO_NOT_TRACK=1`) to disable all CLI and Pi runtime telemetry events.

Adds telemetry visibility to `feynman status`, showing whether telemetry is currently on or off.

Prints a one-time first-run notice on stderr (suppressed in scripts and CI) describing what telemetry collects and how to disable it.

v0.5.0 NOTES ↗ STABLE

Feynman v0.5.0 runs unmodified Pi 0.87.1, passes through Pi session flags, and moves web-search config to ~/.feynman/agent/web-search.json.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.5.0 https://github.com/Companion-
Inc/feynman.git
# already have the repo? check out this version:
$ git checkout v0.5.0
```

└──▷ TRY IT

Resume a previous research session by ID — now that Pi session flags pass through, you can hand off a saved session directly from the CLI.

```
$ feynman --session abc123 "continue the mechanistic
interpretability review"
```

Enable Gemini Web browser-cookie access using the new pi-web-access keys after migrating away from the removed Feynman-specific keys.

~/.feynman/agent/web-search.json

```
{
  "allowBrowserCookies": true,
  "browserCookies": {
    "profile": "Default"
  }
}
```

Run a one-shot research prompt in a CI/headless pipeline — the process now exits cleanly when done instead of hanging.

```
$ feynman --prompt="summarize the top three RLHF alternatives
published in 2024" > rlhf-brief.txt
```

Passes `--continue` / `-c`, `--resume` / `-r`, `--session <path|id>`, `--fork <path|id>`, `--no-session`, and `--export <session.jsonl> [out.html]` through to Pi instead of rejecting them.

Moves web-search configuration to `~/.feynman/agent/web-search.json`, where pi-web-access reads it; an existing `~/.feynman/web-search.json` is migrated automatically on first launch.

Replaces Feynman-specific Gemini Web cookie keys with pi-web-access's `allowBrowserCookies` and `browserCookies.profile` keys in `web-search.json`.

Sets `subagents.agentExcludeDirs` to `["~/.agents"]` so agent files there no longer silently replace Feynman's bundled agents.

Runs Pi 0.87.1 unmodified — no patching, renaming, or sealed `runtime-workspace.tgz` archive — cutting `feynman --help` and `feynman doctor` startup to about one second.

+4 more

└──▷ **BREAKING ON UPGRADE**

- ! The `/feynman-model` slash command is removed; use Pi's `/model` for the main model and pi-subagents' `/subagents` or `subagents.agentOverrides.<name>.model` for per-subagent overrides.
- ! The Feynman-specific `geminiBrowser`, `allowBrowserAuth`, `browserAuth`, and `chromeProfile` keys in `web-search.json` are no longer read; set `allowBrowserCookies` and `browserCookies.profile` in `~/.feynman/agent/web-search.json` instead.
- ! The sealed runtime workspace (`~/.feynman/runtime-workspace.tgz`, its hashes, and its lock) is removed and is no longer verified on launch.
- ! Bundled agents, skills, and theme are no longer copied into `~/.feynman/agent`; copies left by older releases are deleted on first launch unless you edited them.

v0.4.0 NOTES ↗ STABLE

Feynman v0.4.0 adds Semantic Scholar citation-sorted search and OpenAlex semantic search to its literature database tool.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.4.0 https://github.com/Companion-
Inc/feynman.git
# already have the repo? check out this version:
$ git checkout v0.4.0
```

└──▷ **TRY IT**

Verify your alphaXiv session is still valid after a token expiry, especially in a headless CI or SSH environment.

```
$ feynman alpha status
```

Adds `source: "semanticscholar"` to `feynman_science_database_search`, with `sort: "relevance"` and `sort: "pub_date"` options; set `SEMANTIC_SCHOLAR_API_KEY` to use your own rate limit.

Adds `semantic:` prefix to `openalex` queries in `feynman_science_database_search` to invoke OpenAlex embedding search (`search.semantic`) instead of keyword search.

Supports `OPENALEX_API_KEY` to authenticate OpenAlex requests; missing keys now surface 401/403/429 errors with instructions for obtaining a free key.

Adds `feynman alpha status` to check alphaXiv login state against the alphaXiv server.

Routes the researcher agent by source type (Semantic Scholar for general ML/CS, PubMed and Europe PMC for biomedicine, OpenAlex for citation graphs and semantic search) and runs 2–4 reworded queries per question.

+1 more

└──▷ **BREAKING ON UPGRADE**

- ! `feynman serve` (and its `--port`, `--no-open`, `--no-auth`, `--host` flags), `feynman watch`, `feynman jobs`, `feynman rank`, and `feynman paper` are removed; they now exit with status 1.
- ! `feynman_science_database_search` no longer accepts `arxiv_search:` or free-text arXiv queries; arXiv is now ID-only (bare IDs, URLs, or `arxiv_get_papers:`).
- ! The Pi tools `feynman_connector_tools`, `feynman_connector_call`, `feynman_workbench_context`, `feynman_open_chemistry_sketcher`, and `feynman_model_endpoint_call` are removed.
- ! Feynman no longer creates `~/.feynman/active-org.json` or `~/.feynman/orgs/`; existing data in those paths is left in place but no longer managed.
- ! The `/commands` and `/capabilities` slash commands are removed; use `/help` and `/tools` instead.
- ! If you customized `~/.feynman/agent/agents/researcher.md` or `verifier.md`, the new researcher routing and verifier cross-check logic is not applied automatically — you must merge the upstream changes by hand.

v0.3.49 NOTES ↗ STABLE

Feynman v0.3.49 strips PII from telemetry, speeds startup, and ships async subagent workflows via `runs.run` / `runs.all`.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.3.49 https://github.com/Companion-
Inc/feynman.git
# already have the repo? check out this version:
$ git checkout v0.3.49
```

└──▷ **TRY IT**

Opt out of all telemetry before running a research query that includes sensitive prompt text.

```
$ FEYNMAN_TELEMETRY=off feynman "what do we know about adversarial prompt injection"
```

Disables `FEYNMAN_TELEMETRY=off` or `DO_NOT_TRACK=1` to opt out of the new PostHog telemetry project; research-runtime traces no longer carry the command line, host name, or user name.

Speeds up `feynman --version` to answer immediately without first verifying the full bundled research runtime.

Updates pi-subagents to 0.65.1 with async workflow scripts using `runs.run` / `runs.all`, replacing the obsolete top-level `tasks` / `chain` parameters and `/chain` / `/parallel` commands.

Updates web research to pi-web-access 0.28.0 with bounded batch searches, retrievable response IDs, citation preservation, and session-scoped page-answer models.

Moves bundled paper tools to `@companion-ai/alpha-hub@0.1.4`, including the current alphaXiv OAuth/MCP API and result parser.

+3 more

└─▷ **BREAKING ON UPGRADE**

- ! The obsolete top-level `tasks` and `chain` parameters and the `/chain` and `/parallel` commands are removed; subagent workflows must now use the `runs.run` / `runs.all` async workflow interface.
- ! Feynman's own `/thinking` command is removed; use Pi's built-in `/thinking` instead.
- ! The npm package has moved from `@advaitpaliwal/feynman` back to `@companion-ai/feynman`; users of the interim 0.3.48 package must run `npm uninstall -g @advaitpaliwal/feynman` then `npm install -g @companion-ai/feynman`.

WAS THIS USEFUL? 👍 👎

OpenClaw

Feedback ⓘ v2026.9.6 NOTES ↗ CODE ↗ RANK

OpenClaw is a personal AI assistant platform that connects messaging channels, tools, and model providers through a gateway.

> OpenClaw v2026.9.6 adds Lightpanda CDP support to its browser integration, automatic Code Mode selection for preferred models, and readiness checks for the Discord bot.

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

01 **Discord bot readiness checks** NEW 40

Adds Discord bot readiness checks without requiring a QA Lab.

— States what changed and removes a prior requirement. v2026.9.6

02 **Lightpanda CDP support in browser integration** NEW 30

The browser integration now supports Lightpanda connection-scoped CDP replies.

— Names the surface but no mechanism or usage detail. v2026.9.6

03 **Automatic Code Mode for preferred models** NEW 25

OpenClaw now automatically enables Code Mode for preferred models.

— Only mentioned in summary, no mechanism given. v2026.9.6

v2026.9.6 NOTES ↗ CODE ↗ LATEST STABLE

OpenClaw v2026.9.6 adds Lightpanda CDP support, automatic Code Mode for preferred models, and Discord bot readiness checks.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v2026.9.6
https://github.com/openclaw/openclaw.git
# already have the repo? check out this version:
$ git checkout v2026.9.6
```

Supports Lightpanda connection-scoped CDP replies in the browser integration.

Adds Discord bot readiness checks without requiring a QA Lab.

WAS THIS USEFUL?  

AWS Strands Agents

Feedback ⓘ harness-cli/v0.1.2 NOTES ↗

RANK

Strands Agents is AWS's open-source SDK for production agents in Python and TypeScript, with a model-driven loop, tool integration, and deployment targets.

> Strands Agents v0.1.2 adds a scope-based agent cleanup method and ships two breaking changes: an MCP TypeScript client major-version bump and stricter model ID requirements for bidirectional streaming providers.

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

01 **MCP TypeScript client upgraded to 2.0** BREAKING 65

Upgrades the MCP TypeScript integration dependency to `@modelcontextprotocol/client` 2.0; code pinned to the previous major version of that client will need to be updated.

– Names exact dependency and version, with migration note. harness-cli/v0.1.2

THINNER COVERAGE BELOW

02 **Explicit model IDs required for bidi providers** BREAKING 55

Providers using bidirectional streaming (`bidi`) now require an explicit model ID to be supplied; previously a default or implicit ID was accepted.

– States before/after behavior but no code example of the fix. harness-cli/v0.1.2

03 **Agent.shutdown() for scope-based cleanup** NEW 50

Adds Agent.shutdown(), a new method for scope-based cleanup of agents.

– Names exact method but no mechanism detail beyond 'scope-based cleanup'.

harness-cli/v0.1.2

└──▷ **BREAKING ON UPGRADE**

! The MCP TypeScript integration now depends on `@modelcontextprotocol/client` 2.0 — any code pinned to the previous major version of that client will need to be updated.

! Providers using bidirectional streaming now require an explicit model ID to be supplied; previously a default or implicit ID was accepted.

harness-cli/v0.1.2 NOTES ↗ LATEST STABLE

> Strands Agents v0.1.2 adds Agent.shutdown() for scope-based cleanup, upgrades MCP TypeScript client to 2.0, and requires explicit model IDs for bidi providers.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch harness-cli/v0.1.2 https://github.com/strands-agents/harness-sdk.git
# already have the repo? check out this version:
$ git checkout harness-cli/v0.1.2
```

Upgrades the MCP TypeScript client dependency to `@modelcontextprotocol/client` 2.0.

Requires explicit model IDs when using providers with bidirectional streaming (`bidi`).

└──▷ **BREAKING ON UPGRADE**

! The MCP TypeScript integration now depends on `@modelcontextprotocol/client` 2.0 — any code pinned to the previous major version of that client will need to be updated.

! Providers using bidirectional streaming now require an explicit model ID to be supplied; previously a default or implicit ID was accepted.

WAS THIS USEFUL? 👍 👎

Vercel AI SDK

Feedback ⓘ ai@7.0.113 NOTES ↗ RANK

The Vercel AI SDK provides TypeScript APIs for model generation, structured output, tool use, and streaming application interfaces.

> Vercel AI SDK 7.0.113 adds Gemini 3.8 text-to-speech support with structured metadata, per-turn voice controls, and raw audio output formats.

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

01 **Gemini 3.8 TTS support and audio formats** NEW 67

Adds Gemini 3.8 TTS support with structured speech metadata and per-turn speaker and style controls for prebuilt voices, including new Gemini 3.8 speech model IDs in Google and Gateway types. Also supports explicit raw PCM, mu-law, and A-law audio output formats for Gemini speech, preserving native WAV responses without adding a second header and correctly identifying headerless audio formats.

— Describes mechanism and formats but no API/flag names given ai@7.0.113

ai@7.0.113 NOTES ↗ LATEST STABLE

Vercel AI SDK 7.0.113 adds Gemini 3.8 TTS with structured speech metadata, per-turn speaker/style controls, and raw PCM/mu-law/A-law audio output.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch ai@7.0.113 https://github.com/vercel/ai.git
# already have the repo? check out this version:
$ git checkout ai@7.0.113
```

Adds Gemini 3.8 TTS support with structured speech metadata and per-turn speaker and style controls for prebuilt voices, including new Gemini 3.8 speech model IDs in Google and Gateway types.

Supports explicit raw PCM, mu-law, and A-law audio output formats for Gemini speech, preserving native WAV responses without adding a second header and correctly identifying headerless audio formats.

WAS THIS USEFUL? 👍 👎

Mem0 Feedback ⓘ ts-v3.3.0 NOTES ↗ RANK

Mem0 is a memory layer for AI agents that extracts, stores, and retrieves durable facts across sessions through a hosted or self-hosted store.

> Mem0 introduced User Profiles, giving both the TypeScript and Python clients a full set of methods to generate, retrieve, and configure structured, LLM-filled JSON summaries of a user's memories, with async job tracking and idempotent submission.

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

01 **User Profiles in MemoryClient** NEW 88

Adds getProfile(), generateProfile(), getProfileSettings(), updateProfileSettings(), sampleProfiles(), and getProfileJob() to the TypeScript `MemoryClient`, and the equivalent get_profile(), generate_profile(), get_profile_settings(), update_profile_settings(), sample_profiles(), and get_profile_job() to the Python `MemoryClient` and `AsyncMemoryClient`. Profiles are structured JSON summaries of a user, shaped by a per-project JSON Schema and filled by an LLM from that user's stored memories; generation is asynchronous and every job POST carries an `Idempotency-Key` header (passed as `idempotency_key`) so retries with the same key do not start a duplicate job.

Generate a structured user profile from stored memories, then poll for completion — useful for building a personalized summary at session start.

typescript

```typescript
import { MemoryClient } from 'mem0ai';

const client = new MemoryClient({ apiKey: '<your-api-key>' });

// Kick off async profile generation with an idempotency key
const job = await client.generateProfile('alice', {
idempotencyKey: 'alice-profile-2024-06-01' });

// Poll until the job finishes
const result = await client.getProfileJob('alice', job.jobId);
console.log(result);

// Retrieve the finished profile
const profile = await client.getProfile('alice');
console.log(profile);
```

Inspect and update the JSON Schema that controls what fields appear in a user's profile, allowing you to tailor the structure per project.

```typescript
import { MemoryClient } from 'mem0ai';

const client = new MemoryClient({ apiKey: '<your-api-key>' });

// See current profile schema settings
const settings = await client.getProfileSettings();
console.log(settings);

// Update the schema to capture preferred language and timezone
await client.updateProfileSettings({
  schema: {
    preferred_language: { type: 'string' },
    timezone: { type: 'string' }
  }
});
```

— Names every method, both clients, and the idempotency mechanism ts-v3.3.0 v2.2.0

ts-v3.3.0 NOTES ↗ LATEST STABLE

Mem0 TypeScript client gains User Profiles: structured, LLM-generated JSON summaries with async job tracking and idempotent POSTs.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch ts-v3.3.0 https://github.com/mem0ai/mem0.git
# already have the repo? check out this version:
$ git checkout ts-v3.3.0
```

└──▷ USE IT

Generate a structured user profile from stored memories, then poll for completion — useful for building a personalized summary at session start.

```typescript
import { MemoryClient } from 'mem0ai';

const client = new MemoryClient({ apiKey: '<your-api-key>' });

// Kick off async profile generation with an idempotency key
const job = await client.generateProfile('alice', { idempotencyKey:
'alice-profile-2024-06-01' });

// Poll until the job finishes
const result = await client.getProfileJob('alice', job.jobId);
console.log(result);

// Retrieve the finished profile
const profile = await client.getProfile('alice');
console.log(profile);
```

Inspect and update the JSON Schema that controls what fields appear in a user's profile, allowing you to tailor the structure per project.

```typescript
import { MemoryClient } from 'mem0ai';

const client = new MemoryClient({ apiKey: '<your-api-key>' });

// See current profile schema settings
const settings = await client.getProfileSettings();
console.log(settings);

// Update the schema to capture preferred language and timezone
await client.updateProfileSettings({
  schema: {
    preferred_language: { type: 'string' },
    timezone: { type: 'string' }
  }
});
```

Adds getProfile(), generateProfile(), getProfileSettings(), updateProfileSettings(), sampleProfiles(), and getProfileJob() methods to `MemoryClient` for managing per-user structured JSON profiles shaped by a JSON Schema and filled by an LLM from that user's memories.

v2.2.0 NOTES ↗ STABLE

Mem0 v2.2.0 adds User Profiles to MemoryClient with async generation, idempotent job submission, and schema-driven JSON summaries.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v2.2.0 https://github.com/mem0ai/mem0.git
# already have the repo? check out this version:
$ git checkout v2.2.0
```

Adds get_profile(), generate_profile(), get_profile_settings(), update_profile_settings(), sample_profiles(), and get_profile_job() to `MemoryClient` and `AsyncMemoryClient` for structured, LLM-filled user profile management.

Profile generation is asynchronous; every job POST carries an `Idempotency-Key` so callers can pass the same `idempotency_key` on retry without starting a duplicate job.

Profiles are shaped by a JSON Schema configured per project and produce an always-current JSON summary of a user derived from their memories.

WAS THIS USEFUL?  

LangChain LangGraph

Feedback ⓘ cli==0.4.32 NOTES ↗

RANK

Low-level orchestration framework for building, managing, and deploying long-running, stateful AI agents.

> LangGraph's CLI focused entirely on self-hosted deployment ergonomics this window, adding container-image support, agent/environment targeting, listener placement, and clearer agent configuration flags.

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

01 **CLI options for self-hosted deployments and agent targeting**

NEW 69

The CLI adds `--image-uri` to specify a container image URI for self-hosted deployments instead of building locally, and updates `langgraph deploy` to accept `agent_id` and `environment` arguments for targeting specific agents and environments. It also adds listener placement support for self-hosted deployments, and clarifies agent flags while adding support for environment variable defaults in CLI agent configuration.

Deploy a self-hosted LangGraph agent using a pre-built container image rather than building one locally.

```
$ langgraph deploy --image-uri my-registry.example.com/my-agent:latest
```

— Names flags/args; runnable example only for --image-uri cli==0.4.32

cli==0.4.32 NOTES ↗ LATEST STABLE

LangGraph CLI gains `--image-uri` for self-hosted deployments and `agent_id` / `environment` args for `langgraph deploy`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch cli==0.4.32 https://github.com/langchain-ai/langgraph.git
# already have the repo? check out this version:
$ git checkout cli==0.4.32
```

└──▷ TRY IT

Deploy a self-hosted LangGraph agent using a pre-built container image rather than building one locally.

```
$ langgraph deploy --image-uri my-registry.example.com/my-agent:latest
```

Adds `--image-uri` flag to the CLI for specifying a container image URI in self-hosted deployments.

Updates `langgraph deploy` to accept `agent_id` and `environment` arguments for targeting specific agents and environments.

Adds listener placement support for self-hosted deployments via the CLI.

Clarifies agent flags and adds support for environment variable defaults in CLI agent configuration.

WAS THIS USEFUL? 👍 👎

LangChain

Feedback ⓘ langchain-anthropic==1.7.4 NOTES ↗ RANK

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

> LangChain's Anthropic integration now supports changing an agent's available tools mid-conversation via SystemMessage.

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

01 Mid-conversation tool changes via SystemMessage NEW

55

`langchain-anthropic` 1.7.4 supports mid-conversation tool changes delivered via `SystemMessage`, allowing agents to alter available tools dynamically during a running conversation, for both Anthropic and OpenAI models.

— Names mechanism and models but no code example langchain-anthropic==1.7.4

langchain-anthropic==1.7.4 NOTES ↗ LATEST STABLE

langchain-anthropic 1.7.4 enables mid-conversation tool changes via SystemMessage for Anthropic and OpenAI models.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch langchain-anthropic==1.7.4
https://github.com/langchain-ai/langchain.git
# already have the repo? check out this version:
$ git checkout langchain-anthropic==1.7.4
```

WAS THIS USEFUL? 👍 👎

Stacklok Mecatl

Feedback ⓘ v0.0.40 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 v0.0.40 introduces Mecatl Studio, a new web UI with chat, knowledge, schedules and settings workspaces, alongside agent lifecycle hooks for host editors and Jev delegated-model routing.

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

01 **Mecatl Studio web UI with core workspaces** NEW 69

v0.0.40 bootstraps Mecatl Studio as a new web UI (`apps/`) with a chat workspace supporting sessions, streamed runs, and controls; a global search palette and keyboard-shortcuts reference; a knowledge workspace covering skills, learned skills, learning review, and memory consolidation; a schedules workspace with capability-gated inventory, spec mapping, actions, and fires; and a settings workspace with runtime inventory, storage health, and settings management.

— Names each workspace and sub-feature but only UI navigation, no exact commands. v0.0.40

THINNER COVERAGE BELOW

02 **Agent lifecycle hooks from mecatui** NEW 45

Mecatl now emits agent lifecycle hooks from `mecatui` to host editors, allowing editors to react to agent state transitions.

— Names the module but no hook names or API signature given. v0.0.40

03 **Jev delegated-model routing** NEW 10

v0.0.40 introduces Jev delegated-model routing, mentioned in the release summary but not further elaborated in the changelog bullets.

— Named only in passing with no mechanism or usage detail provided. v0.0.40

v0.0.40 NOTES ↗ LATEST STABLE

Mecatl v0.0.40 launches Mecatl Studio web UI with chat, knowledge, schedules, and settings workspaces, plus Jev delegated-model routing and agent lifecycle hooks for host editors.

└──▷ **GET THIS VERSION** ▾

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

Bootstraps Mecatl Studio as a new web UI (`apps/`) with a full chat workspace supporting sessions, streamed runs, and controls.

Adds a global search palette and keyboard-shortcuts reference to Mecatl Studio.

Adds a knowledge workspace to Mecatl Studio covering skills, learned skills, learning review, and memory consolidation.

Adds a schedules workspace to Mecatl Studio with capability-gated inventory, spec mapping, actions, and fires.

Adds a settings workspace to Mecatl Studio with runtime inventory, storage health, and settings management.

+1 more

WAS THIS USEFUL? 👍 👎

deepset Haystack

Feedback ⓘ v3.2.0 NOTES ↗ RANK

Haystack builds language-model applications with modular pipelines for retrieval-augmented generation and agents.

> Haystack v3.2 adds token-budget and summarization controls for long-running Agents, a process-wide unsafe-deserialization override, token-based document splitting, and ships several breaking changes to Toolset combination and tool-result metadata.

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

01 **CompactionHook with SummarizationCompactor for context management**

NEW 85

`CompactionHook` paired with `SummarizationCompactor` keeps long-running Agents within a context window by summarizing old turns with a cheap model instead of dropping them, rather than exhausting quota or hanging. Configured with `context_window`, `compact_at`, `compact_to` on the hook and `min_keep_steps`, `approximate_summary_tokens` on the compactor, and attached via `hooks={"before_llm": [compaction_hook]}`.

Keep long-running Agents within a context window by summarizing old turns with a cheap model instead of dropping them.

```python
from haystack.components.agents import Agent
from haystack.components.generators.chat import
OpenAIResponsesChatGenerator
from haystack.hooks.compaction import CompactionHook,
SummarizationCompactor

compaction_hook = CompactionHook(
    compactor=SummarizationCompactor(
        chat_generator=OpenAIResponsesChatGenerator(model="gpt-
5.4-nano"),
        min_keep_steps=2,
        approximate_summary_tokens=1_024,
    ),
    context_window=400_000,
    compact_at=0.7,
    compact_to=0.4,
)

agent = Agent(
    chat_generator=OpenAIResponsesChatGenerator(model="gpt-5.4"),
    tools=[web_search],
    hooks={"before_llm": [compaction_hook]},
)
```

— Full runnable example naming every config parameter `v3.2.0`

02 **HAYSTACK_UNSAFE_DESERIALIZATION environment variable** NEW

80

New `HAYSTACK_UNSAFE_DESERIALIZATION` environment variable acts as a process-wide equivalent of `unsafe=True` for all deserialization paths: `Pipeline.load`, `Pipeline.loads`, `Pipeline.from_dict`, `Tool.from_dict`, `State.from_dict`, `ConditionalRouter`, and `OutputAdapter`. Truthy values are `1` or `true`, and the setting is frozen after the first deserialization in the process.

— Names the exact env var and every affected deserialization path `v3.2.0`

THINNER COVERAGE BELOW

03 **Custom filters in serialized components require unsafe loading**

55

BREAKING

Serialized `OutputAdapter` and `ConditionalRouter` components containing Jinja `custom_filters` must now be loaded with Pipeline.load(..., unsafe=True) (or the

equivalent `Pipeline.loads` / `Pipeline.from_dict` option).

— Names affected components and the exact load flag needed v3.2.0

---

04 **tool_result_offloaded meta key now holds a list** BREAKING

50

The `tool_result_offloaded` meta key now always holds a list of store references rather than a single reference string; code that reads this key must index into the list.

— Names the meta key and the exact before/after shape v3.2.0

└──▷ **BREAKING ON UPGRADE**

! Serialized `OutputAdapter` and `ConditionalRouter` components containing Jinja `custom_filters` must now be loaded with Pipeline.load(..., unsafe=True) (or the equivalent `Pipeline.loads` / `Pipeline.from_dict` option).

! The `tool_result_offloaded` meta key now always holds a list of store references rather than a single reference string; code that reads this key must index into the list.

! The `+` operator for combining Toolsets has been removed; pass Toolsets as a list instead: Agent(..., tools=[toolset_a, toolset_b]). Pipelines serialized with a `+ -combined` Toolset (referencing the removed `_ToolsetWrapper` class) cannot be loaded and must be recreated in list form.

! Toolset.add() now only accepts a single `Tool`, not another `Toolset`; combine Toolsets via Agent(..., tools=[toolset_a, toolset_b]).

v3.2.0 NOTES ↗ LATEST STABLE

Haystack v3.2 adds SummarizationCompactor, TokenBudgetHook, stop_run control flag, `HAYSTACK_UNSAFE_DESERIALIZATION` env var, and fluent Pipeline building APIs.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v3.2.0 https://github.com/deepset-
ai/haystack.git
# already have the repo? check out this version:
$ git checkout v3.2.0
```

└──▷ **USE IT**

Cap an Agent's total token spend so long-running runs stop cleanly rather than exhausting quota or hanging.

```python
from haystack.components.agents import Agent
from haystack.components.generators.chat import OpenAIChatGenerator
from haystack.hooks.budget import TokenBudgetHook

agent = Agent(
    chat_generator=OpenAIChatGenerator(model="gpt-5.4-nano"),
    tools=[web_search],
    hooks={"before_llm":
[TokenBudgetHook(max_total_tokens=100_000)]},
)
result = agent.run(messages)
print(result["exit_reason"])  # 'text' or 'token_budget_exceeded'
```

Keep long-running Agents within a context window by summarizing old turns with a cheap model instead of dropping them.

```python
from haystack.components.agents import Agent
from haystack.components.generators.chat import
OpenAIResponsesChatGenerator
from haystack.hooks.compaction import CompactionHook,
SummarizationCompactor

compaction_hook = CompactionHook(
    compactor=SummarizationCompactor(
        chat_generator=OpenAIResponsesChatGenerator(model="gpt-5.4-
nano"),
        min_keep_steps=2,
        approximate_summary_tokens=1_024,
    ),
    context_window=400_000,
    compact_at=0.7,
    compact_to=0.4,
)

agent = Agent(
    chat_generator=OpenAIResponsesChatGenerator(model="gpt-5.4"),
    tools=[web_search],
    hooks={"before_llm": [compaction_hook]},
)
```

Split ingested documents by token count to match the context window of modern OpenAI models before indexing.

```python
from haystack.components.preprocessors import DocumentSplitter

splitter = DocumentSplitter(
    split_by="token",
    split_length=512,
    split_overlap=32,
    tokenizer_encoding="o200k_base",
)
```

Adds `HAYSTACK_UNSAFE_DESERIALIZATION` environment variable as a process-wide equivalent of `unsafe=True` for all deserialization paths (`Pipeline.load` , `Pipeline.loads` , `Pipeline.from_dict` , `Tool.from_dict` , `State.from_dict` , `ConditionalRouter` , `OutputAdapter`); truthy values are `1` or `true` ; the setting is frozen after the first deserialization in the process.

BREAKING ON UPGRADE

- ! Serialized `OutputAdapter` and `ConditionalRouter` components containing Jinja `custom_filters` must now be loaded with Pipeline.load(..., unsafe=True) (or the equivalent `Pipeline.loads` / `Pipeline.from_dict` option).
- ! The `tool_result_offloaded` meta key now always holds a list of store references rather than a single reference string; code that reads this key must index into the list.
- ! The `+` operator for combining Toolsets has been removed; pass Toolsets as a list instead: Agent(..., tools=[toolset_a, toolset_b]). Pipelines serialized with a `+ - combined` Toolset (referencing the removed `_ToolsetWrapper` class) cannot be loaded and must be recreated in list form.
- ! Toolset.add() now only accepts a single `Tool` , not another `Toolset` ; combine Toolsets via Agent(..., tools=[toolset_a, toolset_b]).

WAS THIS USEFUL? 👍 👎

Nous Research Hermes

Feedback ⓘ v2026.9.24 NOTES ↗

CODE ↗ RANK

A terminal-native, self-improving AI agent for autonomous coding and tasks, with persistent memory, agent-created skills, and a multi-platform messaging gateway.

> Hermes overhauled its memory system this window — new context files with persistent cross-session memory, fingerprint-gated `/memory approve` writes, and seven external memory backend plugins — while adding a Credential Vault, Computer Use GUI automation, a scriptable `hermes-agent` one-shot runner, and gateway/CLI improvements spanning queue management, per-platform display defaults, backup integrity checks, and auto-restart on update.

└──▷ **WHAT SHIPPED · 21 FEATURES** most completely described first ? what's the number?

01 **hermes-agent one-shot runner** NEW 90

Adds a `hermes-agent` legacy single-query runner with flags `--base-url`, `--max-turns`, `--enabled-toolsets`, `--disabled-toolsets`, `--list-tools`, and `--save-trajectories` for scripted one-shot queries that exit after one turn.

Run a one-shot query against a repo file in CI without launching the full interactive agent.

```
$ hermes-agent --max-turns 5 --enabled-toolsets files,shell --save-trajectories "summarize README.md"
```

— Six named flags plus a runnable CI-style command product docs

02 **Queue management and live work dock** NEW 90

Adds `/queue list`, `/queue rm N`, `/queue edit N`, and `/queue move A B` subcommands to inspect and manipulate the pending prompt queue mid-run, plus `/queue <prompt>` to explicitly queue a follow-up turn from any mode. The live work dock now shows the active `/goal` (active, parked, or paused, with turns used) on its top row and prompts waiting in `/queue` on its bottom rows.

Remove a queued prompt mid-run so it never executes — useful when a follow-up turn becomes irrelevant after seeing intermediate results.

```
$ /queue rm 2
```

Reorder queued prompts so a higher-priority follow-up runs next without discarding anything.

```
$ /queue move 3 1
```

— All subcommands named with runnable examples product docs

03 **External memory backend plugins** NEW 85

Hermes can plug in external memory backends via seven built-in providers — Honcho (dialectic reasoning), OpenViking (tiered retrieval), Mem0 (cloud extraction), Holographic (local SQLite), RetainDB (hybrid search), ByteRover (CLI-based), and Supermemory — for deeper personalization, plus Hindsight (knowledge graphs) installable from the plugin catalog via `hermes plugins install hindsight`.

Install the Hindsight knowledge-graph memory backend from the plugin catalog to enable graph-based personalization in Hermes.

```
$ hermes plugins install hindsight
```

Install a memory provider plugin from the catalog to enable persistent, external memory beyond built-in MEMORY.md and USER.md storage.

```
$ hermes plugins install hindsight
```

— Names all seven providers plus install command product docs

04 **Memory approval gating with fingerprinted entry IDs** BREAKING 80

Memory chunks now get `memory:<source>:<index>:<fingerprint>` IDs enabling exact reference to specific entries, and the new `/memory approve` command lists the full text of every entry a pending write would overwrite or remove; approval is tied to the exact targeted entry and writes are refused if the entry changed after staging. A memory write staged before this fingerprint-pinning feature existed has no verifiable target and is automatically refused — it must be rejected and recreated.

— Names command and ID format with clear before/after mechanism product docs

05 **Per-platform display defaults in cli-config.yaml** BREAKING 80

Introduces per-platform display defaults for `display.tool_progress`, `display.show_reasoning`, `display.busy_ack_detail`, `display.interim_assistant_messages`, and

`display.long_running_notifications` in `cli-config.yaml`. A `cli-config.yaml` copied from an older `cli-config.yaml.example`, or produced by an older `hermes setup`, may include `tool_progress: all` set globally which overrides these new defaults — delete those lines to restore per-platform behaviour.

— Five named config keys with an explicit migration note product docs

06 **Gateway install --no-start-now flag** NEW 75

Adds `hermes gateway install --no-start-now` flag to write the launchd plist without loading it, deferring gateway start to next login or a manual load.

Install the Hermes gateway plist on a headless or shared machine where you want the service to start at next login rather than immediately.

```
$ hermes gateway install --no-start-now
```

— Named flag with runnable command and exact effect product docs

07 **Installation methods for Hermes Agent** NEW 70

Hermes Agent can be installed on Linux, macOS, WSL2, Termux (Android), or native Windows 10/11 via a single `curl -fsSL https://raw.githubusercontent.com/NousResearch/hermes-agent/main/scripts/install.sh | bash` command, or declaratively via a NixOS module with `nix run`.

Install Hermes Agent on a Linux or macOS machine to get a terminal-native autonomous agent up and running in one step

```
$ curl -fsSL
https://raw.githubusercontent.com/NousResearch/hermes-
agent/main/scripts/install.sh | bash
```

— Runnable install command across five platforms product docs

08 **Backup import integrity validation** IMPROVED 70

Backup import now validates archive integrity before writing — it decompresses every member and checks its CRC, printing Error: backup archive is damaged (N member(s) ...) and halting with the Hermes home left untouched if any check fails.

— Exact error text and before/after behaviour, no command product docs

09 **Agent context files and persistent memory** NEW 65

Hermes now uses `SOUL.md`, `USER.md`, `MEMORY.md`, and `AGENTS.md` as agent-facing context files, each with distinct ownership and injection timing, giving the agent

persistent memory across sessions plus session search.

– Names four files and mechanism but no command shown product docs

10 **Auto-restart gateways after update** IMPROVED 60

After a successful update, Hermes automatically restarts all running gateway profiles under the updated home (root and every `profiles/<name>`) so they pick up new code, while leaving gateways from other installs untouched.

– Clear scope and mechanism but no user action needed product docs

THINNER COVERAGE BELOW

11 **Heartbeat prompt scheduling** NEW 55

Supports a `/heartbeat every <interval> <prompt>` command to keep a long-running session alive by re-entering a check prompt whenever the session goes idle.

Keep a long-running deployment session alive by re-entering a check prompt whenever the session goes idle

```
$ /heartbeat every 10m Check the deployment.
```

– Runnable command but no further mechanism described product docs

12 **Model cost lookup via models.dev** IMPROVED 55

Extends model cost lookup to pull vendor list prices from models.dev for first-party API models (OpenAI, xAI, Anthropic, Google, DeepSeek, Xiaomi) not found in Hermes' bundled price table or the provider's `/models` listing.

– Names six vendors and fallback order, no direct action product docs

13 **SDK exports ListRow and ToggleRow** NEW 50

Exports `ListRow` and `ToggleRow` from the SDK so plugin panes can lay out consistently with core Settings.

– Names exported components and their purpose v2026.9.24

14 **Credential Vault for secure sign-in** NEW 45

Adds a Credential Vault that lets the agent sign into sites and fill passwords, payments, and addresses without exposing the underlying secrets to the model.

– Explains what it protects but no setup steps product docs

15 **Recovered replies after gateway crash** IMPROVED 45

The gateway now delivers recovered replies (with a Recovered reply notice) when it was killed after an agent finished a reply but before it was sent, instead of regenerating

the response.

— Describes the failure case and fix but no user step `product docs`

16 **Local model support with no network egress** NEW 40

Hermes supports running entirely local models with no API key or network egress required.

— States capability but no config or command given `product docs`

17 **⌘K pinned session grouping** IMPROVED 40

⌘K session search now groups pinned sessions first under a Pinned heading, ahead of the Sessions group.

— Names the UI change and shortcut, no further detail `v2026.9.24`

18 **Command palette indexes all settings rows** IMPROVED 35

Indexes every hand-built settings row in the command palette, making all Settings rows reachable as command-palette hits.

— States the change but no navigation path given `v2026.9.24`

19 **Computer Use for GUI automation** NEW 30

Adds Computer Use capability for GUI-level automation, letting the agent operate desktop applications directly.

— Named feature but no mechanism or usage path `product docs`

20 **Hermes Desktop multi-instance connection** NEW 25

The Hermes Desktop client can now connect to multiple Hermes instances simultaneously.

— Bare capability statement with no mechanism or steps `product docs`

21 **Bot Screen on hosted desktop images** NEW 25

Ships Bot Screen on hosted `-desktop` images.

— Bare mention with no description of what it does `v2026.9.24`

└──▷ **BREAKING ON UPGRADE**

! A memory write staged before the fingerprint-pinning feature existed has no verifiable target and will be refused automatically — you must reject it and recreate the change.

! A `cli-config.yaml` copied from an older `cli-config.yaml.example`, or produced by an older `hermes setup`, may include `tool_progress: all` set globally, which overrides the new per-platform defaults — delete those lines to restore per-platform behaviour.

v2026.9.24 NOTES ↗ CODE ↗ LATEST STABLE

Hermes v2026.9.24 adds rebindable function-key voice shortcuts, a `composer.dictate` action, Bot Screen on hosted desktop images, and ⌘K pinned-session grouping.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2026.9.24
https://github.com/NousResearch/hermes-agent.git
# already have the repo? check out this version:
$ git checkout v2026.9.24
```

Ships Bot Screen on hosted `-desktop` images.

⌘K session search now groups pinned sessions first under a Pinned heading, ahead of the Sessions group.

Exports `ListRow` and `ToggleRow` from the SDK so plugin panes can lay out consistently with core Settings.

Indexes every hand-built settings row in the command palette, making all Settings rows reachable as command-palette hits.

WAS THIS USEFUL? 👍 👎

AutoGPT

Feedback ⓘ autogpt-platform-beta-v0.8.1 NOTES ↗ RANK

AutoGPT automates multi-step workflows with continuous AI agents that users can build and deploy.

> AutoGPT Platform beta v0.8.1 ships nested workspace folders, a Zapier MCP integration, multi-select copilot prompts, a PostHog-based feature-flag migration, a wave of new LLMs, and an expanded expert marketplace.

└──▷ **WHAT SHIPPED · 11 FEATURES** most completely described first ? what's the number?

01 **FORCE_ALL_FLAGS developer override** NEW 75

Adds the `FORCE_ALL_FLAGS` environment variable to turn every feature flag on locally for development and testing, letting developers exercise flag-gated functionality without toggling each flag individually.

Enable every feature flag at once in a local development environment to test flag-gated functionality without toggling each flag individually.

env

```
FORCE_ALL_FLAGS=true
```

— Named env var with a runnable example command autogpt-platform-beta-v0.8.1

02 **Feature-flag infrastructure migrated to PostHog** IMPROVED 70

Adds a PostHog feature-flag backend behind the existing API, with flag definitions shared through a Redis cache; adds Sentry error reporting that includes the active feature flags recorded by the flag API; and adds a script that re-creates LaunchDarkly flag definitions and targeting rules in PostHog.

— Names backend, cache, error tool and migration script, but no exact commands

autogpt-platform-beta-v0.8.1

03 **Expert marketplace expansion** NEW 60

Adds eight machine-built generalist experts with skills and routines, nine more core experts, and a go-to-market strategist expert (Zara); seeds the skills marketplace from a private skills catalog; and raises the per-expert skill cap to 150.

— Names counts, expert Zara, and the 150 skill cap autogpt-platform-beta-v0.8.1

THINNER COVERAGE BELOW

04 **Nested workspace folders and file picker improvements** NEW 55

Adds nested workspace folders with folder listing for AutoPilot and expert access to the user's own files, folder attach and expert file filter support, nested folder support in the workspace picker and Files page, and Shift-click range selection in the workspace file picker.

— Names the surfaces but only UI-level description, no exact paths

autogpt-platform-beta-v0.8.1

05 **New LLMs added to catalog** NEW 55

Adds Claude Opus 5.5, GPT-6 Sol, and GPT-6 Luna to the LLM catalog, along with Google Gemma 4 31B, Tencent Hy4 Preview, Alibaba Qwen 3.8 Flash, and InclusionAI Ling 3.0 Flash VL via OpenRouter, and Unbiased Pareto.

— Names every model added, no usage mechanism beyond catalog listing

autogpt-platform-beta-v0.8.1

06 **Inline math rendering in chat and markdown** NEW 45

Adds inline `$…$` math rendering in chat replies and markdown artifacts.

— Names exact syntax delimiter but nothing to configure autogpt-platform-beta-v0.8.1

07 **Zapier added to MCP catalog** NEW 40

Adds Zapier to the official MCP catalog, enabling Zapier automation access from agents.

— Names the integration but no setup detail autogpt-platform-beta-v0.8.1

08 **Multi-select questions in copilot ask-user card** NEW 35

Adds multi-select questions in the copilot ask-user card, allowing agents to present checkbox-style prompts to users.

— Explains behaviour but no config or command autogpt-platform-beta-v0.8.1

09 **New analytics views for activation, retention and experiments** NEW 30

Adds activation, retention, unit-economics, and experiment analytics views.

— Names four view types with no further mechanism autogpt-platform-beta-v0.8.1

10 **'Sent from' badge on delegated copilot messages** NEW 25

Adds a Sent from badge on delegated copilot messages to indicate message origin.

— Bare description of a small UI addition autogpt-platform-beta-v0.8.1

11 **Configurable trial capacity and country eligibility** IMPROVED

Makes trial capacity and country eligibility configurable.

— Bare mention with no config key or values named `autogpt-platform-beta-v0.8.1`

`autogpt-platform-beta-v0.8.1` NOTES ↗ LATEST STABLE

AutoGPT Platform beta v0.8.1 adds nested workspace folders, multi-select copilot questions, Zapier MCP, new LLMs, and a `FORCE_ALL_FLAGS` toggle.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch autogpt-platform-beta-v0.8.1
https://github.com/Significant-Gravitas/AutoGPT.git
# already have the repo? check out this version:
$ git checkout autogpt-platform-beta-v0.8.1
```

└──▷ **USE IT**

Enable every feature flag at once in a local development environment to test flag-gated functionality without toggling each flag individually.

.env

```
FORCE_ALL_FLAGS=true
```

Adds `FORCE_ALL_FLAGS` environment variable to turn every feature flag on locally for development and testing.

Adds nested workspace folders with folder listing for AutoPilot and expert access to the user's own files.

Adds folder attach, expert file filter, and nested folder support in the workspace picker and Files page.

Adds Zapier to the official MCP catalog, enabling Zapier automation access from agents.

Adds multi-select questions in the copilot ask-user card, allowing agents to present checkbox-style prompts.

+17 more

WAS THIS USEFUL? 👍 👎

◆ MCP Servers & Tooling

gram

Feedback ⓘ dashboard@0.124.0 NOTES ↗ RANK

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

> Gram shipped a major identity and agent-access overhaul this window: workload JWT-bearer grants for MCP tokens, an Okta Cross App Access readiness area folded into Identity settings, expanded agent API scopes and upstream-account bindings, a new Watchdog findings MCP tool, and dashboard improvements to Risk Events, project switching, and MCP gateway networking.

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

01 Workload JWT-bearer grant for MCP token endpoint NEW

90

Accepts the workload JWT-bearer assertion grant at the MCP token endpoint: a `jwt-bearer` bearer request exchanges a platform-issued identity token for a 15-minute session scoped to the named MCP server; `grant_types_supported` in authorization server metadata lists `jwt-bearer` where the grant is active.

— Full mechanism, grant type, and metadata field named server@2.10.0

02 Okta Cross App Access readiness in Identity settings IMPROVED

90

Adds the `oktaResourceConnections` API for per-server Cross App Access readiness, derived from the upstream authorization server's advertised grants, the recorded AI agent, and confirmed Okta resource connections. Adds the Okta page to organization settings with a connection card, client ID and verification steps, an applications snapshot with on-demand sync, and a Cross App Access readiness checklist with bulk confirmation, then folds it into the Identity page as concern tabs reachable at `identity?tab=sso|provider|applications|cross-app-access`, with `/okta` and `identity?tab=okta` redirecting there.

Navigate directly to the Okta Cross App Access tab on the Identity page after the Okta workspace is folded into concern tabs.

```
$ # Open in browser
https://<your-gram-instance>/identity?tab=cross-app-access
```

— Names API, URLs and tabs; exact navigation given server@2.10.0 dashboard@0.123.0

03 Separate authentication host for per-server MCP OAuth NEW

80

Supports optionally serving the per-server MCP OAuth authorization server on a separate authentication host via the `GRAM_AUTHENTICATION_HOST_URL` environment variable; user session issuers opt in with `use_authentication_host`.

— Exact env var and opt-in flag named `server@2.10.0`

---

04 **User session issuer scope management** BREAKING

80

Organization administrators can move user-session issuers between project and organization scope, or consolidate one issuer into another, with migration preserving clients, sessions, consents, CIMD allowlists, remote-session credentials, and attached resources. MCP servers can select an existing organization- or project-owned user session issuer during creation or from authentication settings. Access tokens bound to a source issuer are rejected by repointed servers after a migration until the client performs a token refresh against the migrated session.

— Names migration scope and breaking token behaviour `server@2.10.0`

---

05 **Agent API scopes for device sync and hook ingestion** NEW

75

Adds `org:device_agent_sync` and `org:hooks_ingest` scopes to the registry and delegated-policy version 2, enabling agent API keys to poll `agent.getPlugins` and send hook events respectively.

— Names exact scopes and methods but no request examples `server@2.10.0`

---

06 **Risk Events filter and Watchdog evidence cards** IMPROVED

75

Risk Events can be filtered by MCP server, with findings from MCP tool calls showing their server, tool, and enforcement outcome even when no chat session is attached. Watchdog evidence cards surface the message behind a finding, with masked secrets/PII, a link to the session transcript scrolled to the finding, and a link to Agent Sessions — visible only to users with the `chat:read` scope.

Filter Risk Events to a specific MCP server to triage tool-call findings without an attached chat session.

> 📍 # In the Risk Events view, use the MCP server filter dropdown to select the target server; # findings display their server name, tool name, and enforcement outcome inline.

— Names scope and UI filter with example, precise workflow `dashboard@0.123.0`

---

07 **Agent device telemetry headers** NEW

70

Attaches `Gram-Device-Serial`, `Gram-Device-Hostname`, and `Gram-Device-Environment` headers from agent requests to request-log wide events as `gram.agent.device.serial`, `gram.agent.device.hostname`, and `gram.agent.device.environment`.

— Exact header and field names given, no usage walkthrough `server@2.10.0`

---

08 **Watchdog findings MCP tool** NEW 70

Adds the `list_watchdog_findings` read-only Platform MCP tool, surfacing rule-level Watchdog alerts, affected users, client surfaces, and fully redacted sampled evidence.

— Named tool and fields but no example call shown `server@2.10.0`

---

09 **Signing service account pinning for identity provider keys** NEW 70

Adds a `GRAM_IDENTITY_PROVIDER_SIGNING_SERVICE_ACCOUNT` setting that pins which GCP service account the signing credential may impersonate, enforced on every key provision, rotation, and client assertion.

— Names exact setting and enforcement points `server@2.10.0`

---

10 **Domain verification prerequisite for SSO setup** NEW 70

Domain verification is now a prerequisite for SSO setup: WorkOS events `organization_domain.verified`, `organization_domain.deleted`, `organization.created`, and `organization.updated` keep verified domains in sync; onboarding status now reports `domain_verified` and `verified_domains`.

— Names events and status fields, no UI steps given `server@2.10.0`

---

11 **Risk engine shadow mode selection** NEW 70

Introduces a `shadow` mode for the `gram-risk-llm-analyzer` feature flag so the fine-tuned risk model scans traffic alongside the legacy engines without enforcing, letting verdicts be compared before full rollout. Risk engine mode is selected per organization via the multivariate `gram-risk-llm-analyzer` flag: `off`, `shadow`, or `llm`.

— Names flag and its values, no rollout command given `server@2.10.0`

---

12 **Agent identity and upstream account bindings** IMPROVED 65

Agents can be attached to owned upstream accounts, with session and grant-generation checks enforced throughout OAuth and tool execution; agent MCP credentials are scoped to live delegable access. Session management now shows agent identities and their attached upstream accounts, preserving agent identity across sessions and usage views.

– Describes mechanism and scope but no named endpoint or flag dashboard@0.123.0

server@2.10.0

THINNER COVERAGE BELOW

13 **Cmd+K project switching** NEW 40

Adds a Projects group to the Cmd+K command palette, letting you jump to or switch projects by name, slug, or ID directly from the palette.

– UI action only, no config or API named dashboard@0.124.0

14 **Agent key creation wizard and MCP server access management** NEW 40

Create agent keys through a server-scoped permission wizard and manage agent access to MCP servers from the dashboard.

– UI path named, no flag or field detail dashboard@0.123.0

15 **Permission-aware MCP setup guidance for non-admin members** NEW 40

Offers permission-aware Platform MCP setup and task prompts to non-admin members from MCP troubleshooting, skill workflows, Plugins, and organization home.

– Names surfaces but no concrete steps or config dashboard@0.123.0

16 **Private network ingress for MCP gateways** NEW 40

MCP gateways can now use private network ingress, configurable from their dashboard settings.

– UI path named, no further mechanism given dashboard@0.123.0

17 **View user profile from sidebar** NEW 30

Adds a View user profile item to the sidebar account menu, opening your own profile.

– Minimal UI addition, no other detail dashboard@0.123.0

└──▷ BREAKING ON UPGRADE

! Access tokens bound to a source issuer are rejected by repointed servers after an issuer migration until the client performs a token refresh against the migrated session.

dashboard@0.124.0 NOTES ↗ LATEST STABLE

gram dashboard 0.124.0 adds Cmd+K project switching by name, slug, or ID.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch dashboard@0.124.0 https://github.com/speakeasy-api/gram.git
# already have the repo? check out this version:
$ git checkout dashboard@0.124.0
```

Adds a Projects group to the Cmd+K palette, letting you jump to or switch projects by name, slug, or ID directly from the command palette.

server@2.10.0 NOTES ↗ STABLE

gram server 2.10.0 adds workload JWT-bearer grants, agent hook ingestion, a Watchdog MCP tool, and Okta Cross App Access readiness.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch server@2.10.0 https://github.com/speakeasy-api/gram.git
# already have the repo? check out this version:
$ git checkout server@2.10.0
```

Adds `org:device_agent_sync` and `org:hooks_ingest` scopes to the registry and delegated-policy version 2, enabling agent API keys to poll `agent.getPlugins` and send hook events respectively.

Adds the `list_watchdog_findings` read-only Platform MCP tool, surfacing rule-level Watchdog alerts, affected users, client surfaces, and fully redacted sampled evidence.

Accepts the workload JWT-bearer assertion grant at the MCP token endpoint: a `jwt-bearer` bearer request exchanges a platform-issued identity token for a 15-minute session scoped to the named MCP server; `grant_types_supported` in authorization server metadata lists `jwt-bearer` where the grant is active.

Adds the `oktaResourceConnections` API for per-server Cross App Access readiness, derived from the upstream authorization server's advertised grants, the recorded AI agent, and confirmed Okta resource connections.

Supports optionally serving the per-server MCP OAuth authorization server on a separate authentication host via the `GRAM_AUTHENTICATION_HOST_URL` environment variable; user session issuers opt in with `use_authentication_host`.

+9 more

└──▷ BREAKING ON UPGRADE

! Access tokens bound to a source issuer are rejected by repointed servers after an issuer migration until the client performs a token refresh against the migrated session.

dashboard@0.123.0 NOTES ↗ STABLE

gram: Risk Events can now be filtered by MCP server, and findings from MCP tool calls show their server, tool, and enforcement outcome even when no chat session is attached.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch dashboard@0.123.0 https://github.com/speakeasy-api/gram.git
# already have the repo? check out this version:
$ git checkout dashboard@0.123.0
```

└──▷ TRY IT

Navigate directly to the Okta Cross App Access tab on the Identity page after the Okta workspace is folded into concern tabs.

```
$ # Open in browser
https://<your-gram-instance>/identity?tab=cross-app-access
```

Filter Risk Events to a specific MCP server to triage tool-call findings without an attached chat session.

📍 # In the Risk Events view, use the MCP server filter dropdown to select the target server; # findings display their server name, tool name, and enforcement outcome inline.

Watchdog evidence cards now surface the message behind a finding, with masked secrets/PII, a link to the session transcript scrolled to the finding, and a link to Agent Sessions — visible only to users with the `chat:read` scope.

Create agent keys through a server-scoped permission wizard and manage agent access to MCP servers.

Show agent identities and their attached upstream accounts in session management.

Adds the Okta page to organization settings, including the connection card, client ID and verification steps, the applications snapshot with on-demand sync, and the Cross App Access readiness checklist with bulk confirmation.

> Scope agent MCP credentials to live delegable access, support agent-owned upstream account bindings, and preserve agent identity in sessions and usage views.
>
> +3 more

WAS THIS USEFUL?  

Prefect FastMCP

Feedback ⓘ v4.0.6 NOTES ↗ RANK

FastMCP is a Python framework for building MCP servers and clients, handling transport, schema generation, authentication, and server composition.

> FastMCP v4.0.6 adds an experimental search-narrowing transform, surfaces filesystem discovery errors, and improves schema introspection for structured tool inputs.

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

01 **FileSystemProvider discovery failure exposure** IMPROVED 45

Exposes `FileSystemProvider` discovery failures so callers can detect and handle them rather than silently losing resources.

– Names the provider and behaviour change but no API detail. v4.0.6

02 **Experimental JevSearchTransform for search narrowing** NEW 43

Adds an experimental `JevSearchTransform` to the transforms subsystem, intended for narrowing search results.

– Names the transform but not its mechanism or usage. v4.0.6

03 **Nested field naming in CodeMode schemas** IMPROVED 35

Names nested fields in detailed `CodeMode` schemas for clearer introspection of structured tool inputs.

– Describes change briefly without mechanism or example. v4.0.6

v4.0.6 NOTES ↗ LATEST STABLE

FastMCP v4.0.6 adds experimental JevSearchTransform, exposes FileSystemProvider discovery failures, and hardens resource template URI matching.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v4.0.6 https://github.com/PrefectHQ/fastmcp.git
# already have the repo? check out this version:
$ git checkout v4.0.6
```

Adds experimental `JevSearchTransform` to the transforms subsystem for narrowing search results.

> Exposes `FileSystemProvider` discovery failures so callers can detect and handle them rather than silently losing resources.
>
> Names nested fields in detailed `CodeMode` schemas for clearer introspection of structured tool inputs.

WAS THIS USEFUL? 👍 👎

MCP TypeScript SDK

Feedback ⓘ

`@modelcontextprotocol/server@2.1.0` NOTES ↗ RANK

The MCP TypeScript SDK provides TypeScript clients, servers, and transports for Model Context Protocol.

> The MCP TypeScript SDK's 2.1.0 releases add DPoP sender-constrained OAuth tokens on the client and per-primitive OAuth scope enforcement on the server, alongside new request-body and JSON-RPC batch size limits, stricter Mcp-Name/protocol-version header validation, and cleaner stdio transport teardown.

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

01 **Per-primitive OAuth scope enforcement** NEW 95

Adds a `scopeChallenge` callback to tool, resource, resource-template, and prompt registrations so servers can enforce fine-grained OAuth scopes at request time, returning HTTP 403 with an `insufficient_scope` WWW-Authenticate challenge before the handler runs. Adds a `requireScopes` helper for static all-of scope checks, and an optional `AuthInfo.resourceMetadataUrl` field stamped by `requireBearerAuth` / `verifyBearerToken`, surfaced via RFC 8707 `resource_metadata` in WWW-Authenticate challenge headers.

Gate a sensitive tool behind a specific OAuth scope, returning a 403 challenge automatically if the caller lacks it.

```typescript
import { McpServer, requireScopes } from
'@modelcontextprotocol/server';
import * as z from 'zod/v4';

const server = new McpServer({ name: 'secure-server', version:
'1.0.0' });

server.registerTool(
    'delete-secret',
    {
        description: 'Delete a stored secret',
        inputSchema: z.object({ key: z.string() }),
        scopeChallenge: requireScopes(['secrets:write'])
    },
    async ({ key }) => ({
        content: [{ type: 'text', text: `Deleted ${key}` }]
    })
);
```

— Names every surface with mechanism and a runnable example.

`@modelcontextprotocol/server@2.1.0`

---

02 **Configurable request body size limits** NEW 95

Adds a `maxRequestBodySize` option (bytes, default `DEFAULT_MAX_REQUEST_BODY_SIZE` = 4 MiB) to `WebStandardStreamableHTTPServerTransportOptions`, `CreateMcpHandlerOptions`, `CreateMcpHonoAppOptions`, `ToNodeHandlerOptions`, and `ToWebRequestOptions`; over-limit bodies are rejected with `413 Payload Too Large` before parsing, or via a `RequestBodyTooLargeError` (`status: 413`) for `toWebRequest` callers unless a pre-parsed `parsedBody` is supplied. Also exports `readRequestBody` for adapter authors needing the SDK's bounded body reader.

Cap inbound request bodies at 512 KiB on a high-traffic Hono deployment to reduce denial-of-service exposure.

```typescript
import { createMcpHandler } from '@modelcontextprotocol/server';
import { createMcpHonoApp } from '@modelcontextprotocol/hono';
import { serve } from '@hono/node-server';

const handler = createMcpHandler(buildServer, {
maxRequestBodySize: 512 * 1024 });
const app = createMcpHonoApp({ maxRequestBodySize: 512 * 1024 });
app.all('/mcp', c => handler.fetch(c.req.raw));
serve({ fetch: app.fetch, port: 3000, hostname: '127.0.0.1' });
```

— Named config keys, default, error path, and example. `@modelcontextprotocol/server@2.1.0`

03 **DPoP sender-constrained access tokens** NEW 95

Adds DPoP (RFC 9449) sender-constrained access token support: implement OAuthClientProvider.dpop() returning a `DpopSession` to opt in, with new exports `generateDpopKeyPair`, `accessTokenHash`, `isDpopNonceChallenge`, and withDpop(session, getToken) for manual token management. `StreamableHTTPClientTransport`, `SSEClientTransport`, and `withOAuth` wrap resource-server `fetch` with the new withDpopFromProvider(provider) middleware, presenting `token_type: 'DPoP'` access tokens as `Authorization: DPoP <token>` plus a fresh per-request proof and retrying `use_dpop_nonce` challenges once. Adds `OAuthErrorCode.InvalidDpopProof` and `OAuthErrorCode.UseDpopNonce`; `extractWWWAuthenticateParams` recognizes the `DPoP` challenge scheme; `OAuthMetadataSchema` gains `dpop_signing_alg_values_supported`.

— Full mechanism and every named surface, but no example.

`@modelcontextprotocol/client@2.1.0`

04 **Mcp-Name header enforcement for task methods** BREAKING 78

Servers now emit and validate the Mcp-Name header on `tasks/get`, `tasks/update`, and `tasks/cancel` requests per SEP-2663, so conforming servers no longer reject task polls with `-32020 HeaderMismatch`. Hand-rolled clients that omit Mcp-Name on these modern-era (2026-07-28 protocol) POSTs now receive `400` / `-32020` (`HeaderMismatch`) and must add the header.

— Names header, methods, error code, and migration fix.

`@modelcontextprotocol/client@2.1.0` `@modelcontextprotocol/server@2.1.0`

05 **Standardized error cause chain across SDK errors** DEPRECATED 75

The `error.data.cause` slot on `SdkError` / `SdkHttpError` is deprecated and slated for removal; read `error.cause` (standard `Error.cause` chain) instead. `SdkError` and `SdkHttpError` now accept standard `ErrorOptions` as an optional fourth constructor argument, forwarding it to `Error` so underlying causes (e.g. `ENOTFOUND`, `ECONNREFUSED`, `ETIMEDOUT`) are reachable via that chain.

— Names deprecated field and its replacement mechanism.

`@modelcontextprotocol/client@2.1.0` `@modelcontextprotocol/server@2.1.0`

---

06 **Expanded Windows env var inheritance for stdio clients**

`IMPROVED` 65

Adds `COMSPEC`, `PATHEXT`, `PROGRAMDATA`, PROGRAMFILES(X86), `PROGRAMW6432`, and `WINDIR` to `DEFAULT_INHERITED_ENV_VARS` so Windows stdio servers spawned by `StdioClientTransport` inherit the environment variables required by PowerShell and Windows OpenSSH.

— Names exact env vars and affected transport, no example.

`@modelcontextprotocol/client@2.1.0`

---

07 **MCP-Protocol-Version header now required** `BREAKING` 60

Hand-rolled clients that omit the MCP-Protocol-Version header on modern (2026-07-28) POSTs now receive `400 Bad Request` / `-32020` (`HeaderMismatch`) and must add the header.

— Clear before/after and exact error, no mechanism beyond that.

`@modelcontextprotocol/server@2.1.0`

---

08 **Clean stdin teardown for StdioServerTransport** `IMPROVED` 60

`StdioServerTransport` now attaches `end` / `close` listeners on stdin so the server process tears down cleanly when the MCP client closes the pipe, with in-flight request handlers receiving `signal.aborted`.

— Behaviour change described but no code sample. `@modelcontextprotocol/server@2.1.0`

THINNER COVERAGE BELOW

---

09 **JSON-RPC batch size capped at 100 messages** `BREAKING` 55

JSON-RPC batch arrays longer than 100 messages are now answered `400` / `-32600` and none of the batch is dispatched.

— Limit and error code named, no further mechanism. `@modelcontextprotocol/server@2.1.0`

└──▷ **BREAKING ON UPGRADE**

- ! Hand-rolled clients that omit the `MCP-Protocol-Version` header on modern (2026-07-28) POSTs now receive `400 Bad Request` / `-32020` (`HeaderMismatch`) — they must add the header.
- ! Hand-rolled clients that omit `Mcp-Name` on `tasks/get`, `tasks/update`, or `tasks/cancel` modern-era POSTs now receive `400` / `-32020` (`HeaderMismatch`) — they must add the header.
- ! Callers of `toWebRequest` that do not pass a pre-parsed body must handle the new rejection (error `name: 'RequestBodyTooLargeError'`, `status: 413`) when the Node stream exceeds the body size limit; pass a `parsedBody` to bypass the SDK read entirely.
- ! The `error.data.cause` slot on `SdkError` / `SdkHttpError` is deprecated and slated for removal; read `error.cause` (standard `Error.cause` chain) instead.
- ! Hand-rolled MCP clients that omit the `Mcp-Name` header on `tasks/get`, `tasks/update`, or `tasks/cancel` requests will now receive `400` / `-32020` (`HeaderMismatch`) from servers built with this SDK when using the 2026-07-28 protocol era.

`@modelcontextprotocol/server@2.1.0` NOTES ↗ LATEST STABLE

Adds per-primitive OAuth scope challenges, configurable request-body size limits, and `Mcp-Name` header enforcement for task methods.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch @modelcontextprotocol/server@2.1.0
https://github.com/modelcontextprotocol/typescript-sdk.git
# already have the repo? check out this version:
$ git checkout @modelcontextprotocol/server@2.1.0
```

└──▷ **USE IT**

Gate a sensitive tool behind a specific OAuth scope, returning a 403 challenge automatically if the caller lacks it.

```typescript
import { McpServer, requireScopes } from
'@modelcontextprotocol/server';
import * as z from 'zod/v4';

const server = new McpServer({ name: 'secure-server', version:
'1.0.0' });

server.registerTool(
    'delete-secret',
    {
        description: 'Delete a stored secret',
        inputSchema: z.object({ key: z.string() }),
        scopeChallenge: requireScopes(['secrets:write'])
    },
    async ({ key }) => ({
        content: [{ type: 'text', text: `Deleted ${key}` }]
    })
);
```

Cap inbound request bodies at 512 KiB on a high-traffic Hono deployment to reduce denial-of-service exposure.

```typescript
import { createMcpHandler } from '@modelcontextprotocol/server';
import { createMcpHonoApp } from '@modelcontextprotocol/hono';
import { serve } from '@hono/node-server';

const handler = createMcpHandler(buildServer, { maxRequestBodySize:
512 * 1024 });
const app = createMcpHonoApp({ maxRequestBodySize: 512 * 1024 });
app.all('/mcp', c => handler.fetch(c.req.raw));
serve({ fetch: app.fetch, port: 3000, hostname: '127.0.0.1' });
```

Adds `scopeChallenge` callback to tool, resource, resource-template, and prompt registrations so servers can enforce fine-grained OAuth scopes at request time, returning HTTP 403 with an `insufficient_scope` `WWW-Authenticate` challenge before the handler runs.

Adds `requireScopes` helper for static all-of OAuth scope checks on registered primitives.

Adds optional `AuthInfo.resourceMetadataUrl` field stamped by `requireBearerAuth` / `verifyBearerToken`, surfaced in `WWW-Authenticate` challenge headers via RFC 8707 `resource_metadata`.

Adds `maxRequestBodySize` option (bytes, default `DEFAULT_MAX_REQUEST_BODY_SIZE` = 4 MiB) to `WebStandardStreamableHTTPServerTransportOptions`, `CreateMcpHandlerOptions`, `CreateMcpHonoAppOptions`, and `ToNodeHandlerOptions` / `ToWebRequestOptions`; over-limit bodies receive `413 Payload Too Large` before parsing.

Exports `readRequestBody` for adapter authors who need the SDK's bounded body reader.

+3 more

└──▷ **BREAKING ON UPGRADE**

- ! Hand-rolled clients that omit the `MCP-Protocol-Version` header on modern (2026-07-28) POSTs now receive `400 Bad Request` / `-32020` (`HeaderMismatch`) — they must add the header.
- ! Hand-rolled clients that omit `Mcp-Name` on `tasks/get`, `tasks/update`, or `tasks/cancel` modern-era POSTs now receive `400` / `-32020` (`HeaderMismatch`) — they must add the header.
- ! Callers of `toWebRequest` that do not pass a pre-parsed body must handle the new rejection (error `name: 'RequestBodyTooLargeError'`, `status: 413`) when the Node stream exceeds the body size limit; pass a `parsedBody` to bypass the SDK read entirely.
- ! The `error.data.cause` slot on `SdkError` / `SdkHttpError` is deprecated and slated for removal; read `error.cause` (standard `Error.cause` chain) instead.

`@modelcontextprotocol/client@2.1.0` NOTES ↗ STABLE

MCP TypeScript client SDK 2.1.0 adds DPoP sender-constrained token support and Mcp-Name header enforcement for task requests

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch @modelcontextprotocol/client@2.1.0
https://github.com/modelcontextprotocol/typescript-sdk.git
# already have the repo? check out this version:
$ git checkout @modelcontextprotocol/client@2.1.0
```

Adds DPoP (RFC 9449) sender-constrained access token support: implement OAuthClientProvider.dpop() returning a `DpopSession` to opt in; new exports include `generateDpopKeyPair`, `accessTokenHash`, `isDpopNonceChallenge`, and withDpop(session, getToken) for callers managing tokens directly.

Wraps resource-server `fetch` in `StreamableHTTPClientTransport`, `SSEClientTransport`, and `withOAuth` with the new withDpopFromProvider(provider) middleware, presenting `token_type: 'DPoP'` access tokens as `Authorization: DPoP <token>` plus a fresh per-request proof, retrying `use_dpop_nonce` challenges once.

Adds `OAuthErrorCode.InvalidDpopProof` and `OAuthErrorCode.UseDpopNonce` error codes; `extractWWWAuthenticateParams` now recognizes the `DPoP` challenge scheme; `OAuthMetadataSchema` gains `dpop_signing_alg_values_supported`.

Adds `COMSPEC`, `PATHEXT`, `PROGRAMDATA`, PROGRAMFILES(X86), `PROGRAMW6432`, and `WINDIR` to `DEFAULT_INHERITED_ENV_VARS` so Windows stdio servers spawned by `StdioClientTransport` inherit the environment variables required by PowerShell and Windows OpenSSH.

`SdkError` and `SdkHttpError` now accept standard `ErrorOptions` as an optional fourth constructor argument, forwarding it to `Error` so the underlying cause (e.g. `ENOTFOUND`, `ECONNREFUSED`, `ETIMEDOUT`) is reachable via the standard `error.cause` chain.

└─▷ **BREAKING ON UPGRADE**

! Hand-rolled MCP clients that omit the `Mcp-Name` header on `tasks/get`, `tasks/update`, or `tasks/cancel` requests will now receive `400` / `-32020` (`HeaderMismatch`) from servers built with this SDK when using the 2026-07-28 protocol era.

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Agentic Infrastructure

Google Gemini Enterprise Agent Platform

Feedback ⓘ changelog · Sep 24, 2026 CHANGELOG ↗

Gemini Enterprise Agent Platform provides Google's managed model training, inference, evaluation, and agent deployment services, formerly offered under Vertex AI.

└──▷ TRY IT

Generate and immediately view an interactive HTML vulnerability report after a scan, so you can triage findings without navigating to the output file manually.

```
$ cm report --format html --open
```

Standardize which directories CodeMender skips during scans across all config files and CLI flags to keep noisy vendor or build paths out of results.

```yaml
scan_config:
  exclude_dirs:
    - vendor
    - node_modules
    - build
    - .git
```

└──▷ BREAKING ON UPGRADE

! Agent Platform SDK for Python 2.0.1 (`google-cloud-agentplatform`) restructures namespaces and decouples the agent surface from `google-cloud-aiplatform` — migration instructions are required; see the Agent Platform SDK for Python version 2.0.1 migration guide.

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.8.71 NOTES ↗ RANK

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

> HyperFrames extended its data-vfx-chain effects pipeline with new compositing kernels, added Gemini TTS and service-account auth to media skills, and improved lint output with precise source locations.

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

01 Line and column locations in lint findings IMPROVED

65

Lint findings now include one-based `line` and `column` fields in JSON output, and CLI output shows `file:line:column` locations when unambiguously identified.

– Names exact fields and output format change clearly v0.8.67

02 Backdrop capture and new kernels in data-vfx-chain NEW

60

The `data-vfx-chain` effects pipeline gains backdrop capture for adjustment-layer effects, plus two new two-source kernels: `luma-matte` and `noise`.

– Names exact kernels and capability but no usage example v0.8.71

THINNER COVERAGE BELOW

03 Gemini TTS and service-account auth in media skills NEW

40

Media skills now support Gemini TTS models and service-account authentication.

– Brief description, no config or usage detail given v0.8.70

v0.8.71 NOTES ↗ LATEST STABLE

HyperFrames v0.8.71 adds backdrop capture and luma-matte/noise two-source kernels to the data-vfx-chain effects pipeline.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.8.71 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.71
```

v0.8.70 NOTES ↗ STABLE

HyperFrames v0.8.70 adds Gemini TTS models and service-account auth to media skills.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.8.70 https://github.com/heygen-
com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.70
```

v0.8.67 NOTES ↗ STABLE

HyperFrames v0.8.67 adds one-based `line` and `column` fields to lint findings in both CLI output and JSON.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.8.67 https://github.com/heygen-
com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.67
```

WAS THIS USEFUL? 👍 👎

NVIDIA Model Optimizer

Feedback ⓘ 0.47.0 NOTES ↗

RANK

NVIDIA Model Optimizer provides quantization, pruning, distillation, and other model-compression techniques for deployment.

> NVIDIA Model Optimizer 0.47.0 adds per-expert weight quantization for Transformer Engine's TEGroupedLinear MoE experts, a new NVFP4 activation-headroom calibration algorithm, MLflow run tracking for PTQ scripts, and Aumann-Shapley sensitivity scoring, alongside a wide set of new PTQ/QAD recipes for models such as Kimi-K3, Step-3.7, Qwen3-VL/3.5 and Alpamayo. The release also removes a long list of deprecated flags, config forms and example paths, and renames checkpoint-mirror recipe paths to a canonical Hugging Face Hub id scheme.

└──▷ **WHAT SHIPPED · 26 FEATURES** most completely described first ? what's the number?

01 **NVFP4 activation headroom calibration** NEW 90

Adds the `nvfp4_act_headroom` calibration algorithm for NVFP4 activation global scales, shipping `modelopt_recipes/general/ptq/nvfp4_act_headroom-kv_fp8_cast.yaml`. Configurable via `anchor_percentile` (default 1), `upper_percentile` (default 99.99), and `rho` (default 16384), with a nested `weight_scale_algorithm` supporting `max`, `mse`, or `local_hessian`.

Use the new `nvfp4_act_headroom` calibration recipe to leave headroom above the largest calibration activations, reducing saturation on outlier inputs at inference time.

```
$ python examples/hf_ptq/hf_ptq.py --model_name meta-llama/Llama-3.1-8B-Instruct --recipe modelopt_recipes/general/ptq/nvfp4_act_headroom-kv_fp8_cast.yaml --output_dir ./checkpoints/llama31-act-headroom
```

— Recipe path, defaults, and runnable example given. 0.47.0

02 **MLflow run tracking for PTQ scripts** NEW 90

Adds `--mlflow <tracking-uri>` to `examples/hf_ptq/hf_ptq.py` (also honours `MLFLOW_TRACKING_URI`) to record the invocation, resolved recipe, run log, and quantization summaries as searchable params; experiment name defaults to `$USER/hf_ptq/<checkpoint basename>-<recipe name or --qformat>` and can be overridden with `--mlflow_experiment` / `--mlflow_run_name`. Adds the same `--mlflow <tracking-uri>` support to `examples/vllm_serve/vllm_serve_fakequant.py`, uploading the launcher

command, resolved `RECIPE_PATH`, worker log, and quantizer summary, with experiment name defaulting to `$USER/vllm_serve_fakequant/<model basename>-<recipe name or quantization config>` and overridable via `--mlflow-experiment` / `--mlflow-run-name`. Both are backed by the new `modelopt.torch.utils.mlflow.MlflowRunLogger` utility.

Track an NVFP4 PTQ run in MLflow so the recipe, quantization summaries, and full log are searchable alongside every CLI argument.

```
$ python examples/hf_ptq/hf_ptq.py --model_name meta-llama/Llama-3.1-8B-Instruct --recipe modelopt_recipes/general/ptq/nvfp4_default-kv_fp8_cast.yaml --mlflow https://mlflow.example.com --mlflow_experiment myteam/hf_ptq/llama31-nvfp4
```

— Two flags, defaults, and utility class all named with example. `0.47.0`

03 **Per-expert weight quantization for TEGroupedLinear** BREAKING

90

Adds per-expert weight quantization for Transformer Engine `TEGroupedLinear` (fused MoE experts): each expert gets its own `GroupedQuantizer` holding one `TensorQuantizer` per expert with an independent `amax`, applied across `mtq.quantize` calibration, HF/Megatron export, and QAD. Adds the `MODELOPT_TEGROUPED_COMPILE_WEIGHT_LOOP=1` environment variable to enable opt-in `torch.compile` execution for these per-expert weight quantizers while preserving native checkpoint amax shapes. ModelOpt checkpoints containing quantized `TEGroupedLinear` modules saved before 0.47 are not compatible with 0.47; re-run PTQ to regenerate compatible checkpoints.

Enable `torch.compile` for per-expert weight quantization loops in TEGroupedLinear before running PTQ, reducing quantization overhead on large MoE models.

```
$ MODELOPT_TEGROUPED_COMPILE_WEIGHT_LOOP=1 python examples/hf_ptq/hf_ptq.py --model_name stepfun-ai/Step-3.7-Flash --recipe huggingface/step3p7/ptq/nvfp4_experts_only-kv_fp8_cast --output_dir ./checkpoints/step37-nvfp4
```

— Mechanism, env var, and checkpoint-compatibility break all specified. `0.47.0`

04 **Checkpoint-mirror recipe path renaming** BREAKING

85

Checkpoint-mirror recipe paths move from `huggingface/models/<org>/<checkpoint>/` to `models/<org>/<model_id>/`

keyed by the canonical Hugging Face Hub id — e.g. Step 3.5 Flash moves to `models/stepfun-ai/Step-3.5-Flash/ptq/` and NVIDIA Nemotron recipes gain the `NVIDIA-` prefix. The Mistral Medium 3.5 recipe similarly moves from `huggingface/models/nvidia/Mistral-Medium-3.5-128B-NVFP4/ptq/nvfp4-max-calib` to `models/mistralai/Mistral-Medium-3.5-128B/ptq/nvfp4-max-calib`. Update any saved `--recipe` paths.

— Exact old and new paths given for two checkpoints. `0.47.0`

---

05 **AutoQuantize CLI flags removed** BREAKING 85

The `--auto_quantize_bits`, `--auto_quantize_method`, `--auto_quantize_score_size`, `--auto_quantize_cost_model`, and `--auto_quantize_active_moe_expert_ratio` flags are removed from `examples/hf_ptq` (deprecated in 0.46). Use an AutoQuantize `--recipe` from `modelopt_recipes/general/auto_quantize/` instead; existing `--auto_quantize_checkpoint` files from earlier releases are rejected with `Use a different checkpoint path`.

— All removed flags and the replacement path are named. `0.47.0`

---

06 **FP32 master weights for DFlash training** BREAKING 83

`dflash_fp32_master_weights` now defaults to `True` for all DFlash-family recipes (set to `False` to reclaim optimizer memory). Adds `MasterWeightAdamW`, which holds an fp32 master copy plus fp32 Adam moments, applies updates in fp32, and writes results back at the parameter's dtype for standard mixed-precision placement; and `VerifyMasterWeightsCallback`, which raises after the first training step if `MasterWeightAdamW` was not used, preventing the flag from being silently absent for a whole job.

— Mechanism and default-change impact fully described. `product docs`

---

07 **Legacy qformat aliases removed** BREAKING 80

The `--qformat` / `--quant_cfg` short names `int8_sq`, `int8_wo`, `w4a8_awq`, `nvfp4_awq`, `nvfp4_mse`, `nvfp4_local_hessian`, `fp8_pb_wo`, and `fp8_pc_pt` are removed (deprecated in 0.45). Use the preset basenames under `modelopt_recipes/configs/ptq/presets/model/` instead; `modelopt.recipe.presets.QFORMAT_ALIASES` and the `aliases` argument of load_quant_cfg_choices() are also removed.

— Every removed alias and replacement path named. `0.47.0`

---

08 **Alpamayo QAD training example** NEW 78

Adds `examples/alpamayo/qad.py` for quantization-aware distillation on the quantized Alpamayo checkpoint via `QADTrainer`, with FSDP2 multi-GPU support and `--`

`export` to reassemble a full `AlpamayoR1` checkpoint loadable with `AlpamayoR1.from_pretrained`.

— Script path, class, and flag named, no runnable command given. `0.47.0`

09 **HF export for quantized Qwen3-VL/Qwen3.5-VL** NEW 76

Adds HuggingFace unified export of quantized Qwen3-VL and Qwen3.5-VL checkpoints (PTQ or QAD) via `examples/megatron_bridge/export_quantized_megatron_to_hf.py`, with Qwen3.5-VL also covering GatedDeltaNet linear-attention layers and MoE shared experts.

— Script path and covered layer types named, no example command. `0.47.0`

10 **Megatron-Bridge MoE expert layout auto-selection** IMPROVED 76

Megatron-Bridge scripts now automatically choose the MoE expert layout from model config — `TEGroupedMLP` (grouped GEMM) when exportable to HuggingFace, otherwise `SequentialMLP`; `--no_moe_grouped_gemm` forces the sequential path explicitly.

— Layout classes and override flag named, no example. `0.47.0`

11 **Legacy layerwise config form removed** BREAKING 76

The legacy `layerwise` bool form, its `use_sequential` alias, and the top-level `layerwise_checkpoint_dir` key are removed from calibration algorithm configs (deprecated in 0.45). Use the nested form `layerwise: {enable: true, checkpoint_dir: /path}`; a pre-0.45 `modelopt_state` carrying either legacy key now fails validation on restore instead of being migrated.

— Old and new config forms and restore-failure behaviour given. `0.47.0`

12 **SFT-masked distillation flags** NEW 73

Adds `--sft` and `--sft_dataset_root <dir>` flags to `examples/megatron_bridge/distill.py` for SFT-masked distillation on raw prompt-completion JSONL (`{"input", "output"}` records) with loss masked to response tokens.

— Flags and data format named, no runnable example. `0.47.0`

13 **Kimi-K3 calibration-free converter** NEW 73

Adds a calibration-free streaming Kimi-K3 converter and checkpoint-mirror recipe for NVFP4 routed experts with `input_scale=1.0` and 128x128 block-FP8 KDA/MLA attention weights, operating shard-by-shard on packed MXFP4 experts.

— Mechanism and parameters named, no example command. `0.47.0`

14 **FP8 Vision Encoder recipes for Qwen3-VL/3.5** NEW 73

Adds opt-in FP8 Vision Encoder recipes under the `qwen3_vl` and `qwen3_5` model types: a vision-only recipe (LM and KV cache stay in high precision) and a joint recipe (Vision Encoder + LM Linears + FP8 KV-cache cast), both quantizing primary and deepstack merger Linears while leaving patch embedding and vision-attention BMMs in high precision.

— Model types and quantized/unquantized layers named, no exact recipe path. 0.47.0

15 **Multimodal calibration forwarding change** BREAKING 73

Image-text calibration with `--calib_with_images` now forwards multimodal batches through the complete VLM for all VLM families, so existing non-Nemotron commands may produce different language-model activation ranges and output scales. Recipe-based VLM PTQ also targets the complete VLM; vision modules stay in high precision by default, so custom recipes must explicitly exclude vision modules when required.

— Flag and behavioural before/after described clearly. 0.47.0

16 **clamp_kv_cache_scales export flag** NEW 70

Adds `clamp_kv_cache_scales` to `export_mcore_gpt_to_hf`; set to `False` to preserve learned FP8 KV-cache scales from a QAT Megatron-Core model (default keeps the existing minimum scale of 1.0).

— Flag, default, and effect stated; no example command. 0.47.0

17 **Muse Glimmer AutoQuantize recipe** NEW 70

Adds a Muse Glimmer AutoQuantize recipe that searches language-model MLP projections, self-attention projections, and `lm_head` over W4A16 NVFP4 Four-Over-Six, FP8, and BF16 fallback at 5.5 effective bits while leaving the vision tower unquantized.

— Search space and bit target named, no example command. 0.47.0

18 **Step-3.7 PTQ recipes** NEW 68

Adds PTQ support for Step-3.7 (`stepfun-ai/Step-3.7-Flash`) routed experts via new recipes `huggingface/step3p7/ptq/nvfp4_experts_only-kv_fp8_cast` and `huggingface/step3p7/ptq/nvfp4_mlp_only-kv_fp8` .

— Recipe paths given, no usage example shown. 0.47.0

19 **ONNX Autotune runtime-precision benchmarking** IMPROVED 68

ONNX quantization with Autotune now benchmarks placements in the requested runtime precision and retains calibrated INT8/FP8 Q/DQ only when it meets the

configured TensorRT speedup threshold (1.02x by default), otherwise saving the high-precision no-Q/DQ model.

— Mechanism and default threshold given, no config key shown. `0.47.0`

20 **Legacy PTQ example paths removed** BREAKING 66

The `examples/llm_ptq` symlink and `examples/vlm_ptq` forwarder are removed (deprecated in 0.46). Use `examples/hf_ptq`, passing `--vlm` for vision-language models.

— Removed paths and replacement flag stated exactly. `0.47.0`

21 **FAR3D ONNX PTQ example migration** BREAKING 65

The FAR3D ONNX PTQ example is migrated to `quantize_vovnet.py`; decoder calibration, quantization, and related CLI flags are removed — the decoder remains in mixed FP16/FP32 precision only.

— New script name and dropped scope stated, no migration steps. `0.47.0`

22 **In-trainer quant_cfg removed** BREAKING 65

In-trainer quantization via `QuantizationArguments.quant_cfg` / `--quant_cfg` is removed (deprecated in 0.45); use `--recipe`. The `QuantizationArgumentsWithConfig` alias in `modelopt.torch.quantization.plugins.transformers_trainer` is also removed; use `QuantizationArguments`.

— Removed attribute and alias plus replacements named. `0.47.0`

23 **Aumann-Shapley sensitivity scoring** NEW 60

Adds `method='aumann_shapley'` to sensitivity scoring for label-free path-integral sensitivity scoring and predicted calibration damage, with method-specific settings such as an optional damage bound passed in a method dictionary.

— Named method value given, no usage example. `product docs`

24 **iq2_xxs weight-only quantization** NEW 60

Adds `iq2_xxs` weight-only quantization with a CUDA encoder and a `general/ptq` recipe running at 2.0625 bits per weight.

— Recipe and bit-width given, no runnable example. `product docs`

THINNER COVERAGE BELOW

25 **PETR ONNX PTQ examples** NEW 52

Adds end-to-end PETRv1 and PETRv2 ONNX PTQ examples covering calibration, INT8 and FP8 VoVNet backbone quantization, TensorRT deployment, and accuracy evaluation.

— Scope named but no flags, paths or example. 0.47.0

---

26 **Quantizer state utility context managers** NEW 50

Adds `mtq.temporarily_fold_weights` for repeated frozen-weight inference and `mtq.preserve_quantizer_attributes_context` for restoring temporary quantizer property and type changes.

— Names two functions, no usage detail beyond purpose. 0.47.0

└──▷ BREAKING ON UPGRADE

! `dflash_fp32_master_weights` now defaults to `True`; setups relying on the previous default (False) will use more optimizer memory on upgrade.

! The FAR3D ONNX PTQ example is migrated to `quantize_vovnet.py`; decoder calibration, quantization, and related CLI flags are removed — the decoder remains in mixed FP16/FP32 precision only.

! Image-text calibration with `--calib_with_images` now forwards multimodal batches through the complete VLM for all VLM families, so existing non-Nemotron commands may produce different language-model activation ranges and output scales. Recipe-based VLM PTQ also targets the complete VLM; vision modules stay in high precision by default, so custom recipes must explicitly exclude vision modules when required.

! Checkpoint-mirror recipe paths move from `huggingface/models/<org>/<checkpoint>/` to `models/<org>/<model_id>/` keyed by the canonical Hugging Face Hub id (e.g. Step 3.5 Flash moves to `models/stepfun-ai/Step-3.5-Flash/ptq/`; NVIDIA Nemotron recipes gain the `NVIDIA-` prefix). Update any saved `--recipe` paths for these checkpoint recipes.

! The Mistral Medium 3.5 checkpoint-mirror recipe moves from `huggingface/models/nvidia/Mistral-Medium-3.5-128B-NVFP4/ptq/nvfp4-max-calib` to `models/mistralai/Mistral-Medium-3.5-128B/ptq/nvfp4-max-calib`. Update any saved `--recipe` paths.

! The `--auto_quantize_bits`, `--auto_quantize_method`, `--auto_quantize_score_size`, `--auto_quantize_cost_model`, and `--auto_quantize_active_moe_expert_ratio` flags are removed from `examples/hf_ptq` (deprecated in 0.46). Use an AutoQuantize `--recipe` from `modelopt_recipes/general/auto_quantize/` instead. Existing `--auto_quantize_checkpoint` files from earlier releases are rejected with `Use a different checkpoint path`.

! The `examples/llm_ptq` symlink and `examples/vlm_ptq` forwarder are removed (deprecated in 0.46). Use `examples/hf_ptq`, passing `--vlm` for vision-language models.

! The `--qformat` / `--quant_cfg` short names `int8_sq`, `int8_wo`, `w4a8_awq`, `nvfp4_awq`, `nvfp4_mse`, `nvfp4_local_hessian`, `fp8_pb_wo`, and `fp8_pc_pt` are removed (deprecated in 0.45). Use the preset basenames under `modelopt_recipes/configs/ptq/presets/model/` instead. `modelopt.recipe.presets.QFORMAT_ALIASES` and the `aliases` argument of load_quant_cfg_choices() are also removed.

! The legacy `layerwise` bool form, its `use_sequential` alias, and the top-level `layerwise_checkpoint_dir` key are removed from calibration algorithm configs (deprecated in 0.45). Use the nested form `layerwise: {enable: true, checkpoint_dir: /path}`. A pre-0.45 `modelopt_state` carrying either legacy key now fails validation on restore instead of being migrated.

! In-trainer quantization via `QuantizationArguments.quant_cfg` / `--quant_cfg` is removed (deprecated in 0.45); use `--recipe`. The `QuantizationArgumentsWithConfig` alias in `modelopt.torch.quantization.plugins.transformers_trainer` is also removed; use `QuantizationArguments`.

! Transformer Engine `TEGroupedLinear` now uses per-expert weight quantization (one `amax` per expert). ModelOpt checkpoints containing quantized `TEGroupedLinear` modules saved before 0.47 are not compatible with 0.47; re-run PTQ to regenerate compatible checkpoints.

0.47.0 NOTES ↗ LATEST STABLE

NVIDIA Model Optimizer 0.47.0 adds NVFP4 activation calibration, MLflow run tracking, per-expert MoE quantization, and new QAD/PTQ recipes.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch 0.47.0 https://github.com/NVIDIA/Model-Optimizer.git
# already have the repo? check out this version:
$ git checkout 0.47.0
```

└──▷ **TRY IT**

Track an NVFP4 PTQ run in MLflow so the recipe, quantization summaries, and full log are searchable alongside every CLI argument.

```
$ python examples/hf_ptq/hf_ptq.py --model_name meta-llama/Llama-3.1-8B-Instruct --recipe modelopt_recipes/general/ptq/nvfp4_default-kv_fp8_cast.yaml --mlflow https://mlflow.example.com --mlflow_experiment myteam/hf_ptq/llama31-nvfp4
```

Use the new `nvfp4_act_headroom` calibration recipe to leave headroom above the largest calibration activations, reducing saturation on outlier inputs at inference time.

```
$ python examples/hf_ptq/hf_ptq.py --model_name meta-llama/Llama-3.1-8B-Instruct --recipe modelopt_recipes/general/ptq/nvfp4_act_headroom-kv_fp8_cast.yaml --output_dir ./checkpoints/llama31-act-headroom
```

Enable `torch.compile` for per-expert weight quantization loops in TEGroupedLinear before running PTQ, reducing quantization overhead on large MoE models.

```
$ MODELOPT_TEGROUPED_COMPILE_WEIGHT_LOOP=1 python
examples/hf_ptq/hf_ptq.py --model_name stepfun-ai/Step-3.7-Flash --
recipe huggingface/step3p7/ptq/nvfp4_experts_only-kv_fp8_cast --
output_dir ./checkpoints/step37-nvfp4
```

Adds `mtq.temporarily_fold_weights` for repeated frozen-weight inference and `mtq.preserve_quantizer_attributes_context` for restoring temporary quantizer property and type changes.

Adds the `nvfp4_act_headroom` calibration algorithm for NVFP4 activation global scales, shipping `modelopt_recipes/general/ptq/nvfp4_act_headroom-kv_fp8_cast.yaml`; configurable via `anchor_percentile` (default 1), `upper_percentile` (default 99.99), and `rho` (default 16384), with a nested `weight_scale_algorithm` supporting `max`, `mse`, or `local_hessian`.

Adds `--mlflow <tracking-uri>` to `examples/hf_ptq/hf_ptq.py` (also honours `MLFLOW_TRACKING_URI`) to record the invocation, resolved recipe, run log, and quantization summaries as searchable params; experiment name defaults to `$USER/hf_ptq/<checkpoint basename>-<recipe name or --qformat>` and can be overridden with `--mlflow_experiment` / `--mlflow_run_name`.

Adds `--mlflow <tracking-uri>` to `examples/vllm_serve/vllm_serve_fakequant.py` (also honours `MLFLOW_TRACKING_URI`), uploading the launcher command, resolved `RECIPE_PATH`, worker log, and quantizer summary; experiment name defaults to `$USER/vllm_serve_fakequant/<model basename>-<recipe name or quantization config>` and overridable via `--mlflow-experiment` / `--mlflow-run-name`.

Adds `clamp_kv_cache_scales` to `export_mcore_gpt_to_hf`; set to `False` to preserve learned FP8 KV-cache scales from a QAT Megatron-Core model (default keeps the existing minimum scale of 1.0).

+13 more

└──▷ **BREAKING ON UPGRADE**

- ! The FAR3D ONNX PTQ example is migrated to `quantize_vovnet.py`; decoder calibration, quantization, and related CLI flags are removed — the decoder remains in mixed FP16/FP32 precision only.
- ! Image-text calibration with `--calib_with_images` now forwards multimodal batches through the complete VLM for all VLM families, so existing non-Nemotron commands may produce different language-model activation ranges and output

scales. Recipe-based VLM PTQ also targets the complete VLM; vision modules stay in high precision by default, so custom recipes must explicitly exclude vision modules when required.

! Checkpoint-mirror recipe paths move from `huggingface/models/<org>/<checkpoint>/` to `models/<org>/<model_id>/` keyed by the canonical Hugging Face Hub id (e.g. Step 3.5 Flash moves to `models/stepfun-ai/Step-3.5-Flash/ptq/`; NVIDIA Nemotron recipes gain the `NVIDIA-` prefix). Update any saved `--recipe` paths for these checkpoint recipes.

! The Mistral Medium 3.5 checkpoint-mirror recipe moves from `huggingface/models/nvidia/Mistral-Medium-3.5-128B-NVFP4/ptq/nvfp4-max-calib` to `models/mistralai/Mistral-Medium-3.5-128B/ptq/nvfp4-max-calib`. Update any saved `--recipe` paths.

! The `--auto_quantize_bits`, `--auto_quantize_method`, `--auto_quantize_score_size`, `--auto_quantize_cost_model`, and `--auto_quantize_active_moe_expert_ratio` flags are removed from `examples/hf_ptq` (deprecated in 0.46). Use an AutoQuantize `--recipe` from `modelopt_recipes/general/auto_quantize/` instead. Existing `--auto_quantize_checkpoint` files from earlier releases are rejected with `Use a different checkpoint path`.

! The `examples/llm_ptq` symlink and `examples/vlm_ptq` forwarder are removed (deprecated in 0.46). Use `examples/hf_ptq`, passing `--vlm` for vision-language models.

! The `--qformat` / `--quant_cfg` short names `int8_sq`, `int8_wo`, `w4a8_awq`, `nvfp4_awq`, `nvfp4_mse`, `nvfp4_local_hessian`, `fp8_pb_wo`, and `fp8_pc_pt` are removed (deprecated in 0.45). Use the preset basenames under `modelopt_recipes/configs/ptq/presets/model/` instead. `modelopt.recipe.presets.QFORMAT_ALIASES` and the `aliases` argument of load_quant_cfg_choices() are also removed.

! The legacy `layerwise` bool form, its `use_sequential` alias, and the top-level `layerwise_checkpoint_dir` key are removed from calibration algorithm configs (deprecated in 0.45). Use the nested form `layerwise: {enable: true, checkpoint_dir: /path}`. A pre-0.45 `modelopt_state` carrying either legacy key now fails validation on restore instead of being migrated.

! In-trainer quantization via `QuantizationArguments.quant_cfg` / `--quant_cfg` is removed (deprecated in 0.45); use `--recipe`. The `QuantizationArgumentsWithConfig` alias in `modelopt.torch.quantization.plugins.transformers_trainer` is also removed; use `QuantizationArguments`.

! Transformer Engine `TEGroupedLinear` now uses per-expert weight quantization (one `amax` per expert). ModelOpt checkpoints containing quantized `TEGroupedLinear` modules saved before 0.47 are not compatible with 0.47; re-run PTQ to regenerate compatible checkpoints.

WAS THIS USEFUL? 👍 👎

PyTorch ExecuTorch

Feedback ⓘ v1.5.1 NOTES ↗ RANK

ExecuTorch runs PyTorch models on mobile, embedded, and edge devices through an ahead-of-time export path and a lightweight C++ runtime.

> ExecuTorch v1.5.1 adds a C++ SDK with linkable delegate libraries, an opt-in ROCm backend for the AOTI CUDA path, and broad new operator, model, and platform coverage spanning the Arm/Cortex-M/Ethos-U, Qualcomm, Core ML/MLX/Metal, Vulkan, and NXP Neutron backends.

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

01 **Arm, Cortex-M and Ethos-U backend expansion** IMPROVED 75

Adds opt-in Cortex-M explicit-layout lowering with convolution and pooling kernels; Arm export-recipe support, a pre-decomposition partitioner pipeline, and broader symbolic-shape and dim-order handling; Ethos-U65 support and Ethos-U85 KV-cache export; SmolLM2, TinyStories-42M, Silero VAD, MobileSAM, and Stable Diffusion workflows for Arm; upgrades CMSIS-NN to v8.0.0 with aligned Arm runner dependencies; and expands Arm support for symbolic shapes and dynamic convolution and pooling.

— Names every model, chip target and library version v1.5.1

02 **Core ML, MLX and Metal backend expansion** IMPROVED 75

Adds linkable Core ML and MLX libraries to Apple frameworks and SwiftPM plus ETDump profiling support; adds MLX off-graph KV-cache flat, ring, and cell layouts with shared pools; expands MLX lowering with fused attention, `native_group_norm`, `upsample_nearest2d`, `flip`, and `trunc` support; and adds Metal `addmm` and `split_copy` support.

— Names every op and layout added across three backends v1.5.1

03 **Quantized LLM kernel additions** NEW 75

Adds a quantized MoE feed-forward operator with INT4 packing, fused projections, and source transformations; a quantized stacked-halves RoPE operator with BF16-by-FP32 GEMV decode support; and shared INT4, INT5, and INT6 short-query kernels for cross-architecture CUDA inference.

— Names kernel types and precisions, no benchmark numbers v1.5.1

04 **ROCm support in AOTI CUDA backend** NEW 70

Adds opt-in ROCm support to the AOTI CUDA backend, including packed W4 execution and Voxtral Realtime coverage on gfx1100 and gfx950.

— Names GPU targets and workload, no setup steps v1.5.1

05 **Qualcomm backend operator and model coverage** IMPROVED

70

Adds LLM quantization-aware distillation and Gemma 4 text-model enablement for the Qualcomm backend, plus support for `pdist`, `sort`, `reflection_pad3d`, `as_strided`, `scatter_reduce`, `scatter_add`, and Hadamard-transform flows.

— Lists every op and model added, no benchmarks given v1.5.1

06 **Vulkan backend expansion** IMPROVED

70

Adds Vulkan external image and buffer wrapping with caller-provided dimensions, plus payload-bounded constant sharding, scalar comparisons, and broader `arange`, `clamp`, indexing, and batch support.

— Names ops and mechanisms, no usage example v1.5.1

07 **NXP eIQ Neutron backend update** IMPROVED

70

Upgrades the NXP eIQ Neutron SDK to 3.2.1 and adds an MCUXpresso example build, plus Neutron support for `rsqrt`, Hardswish, `softmax`, reflected padding, scalar elementary arguments, and AdaptiveAvgPool1d.

— Names SDK version and every op added v1.5.1

08 **C++ SDK bundled in Python wheel** NEW

65

The Python wheel now ships a C++ SDK with linkable libraries for the CUDA, Core ML, MLX, OpenVINO, and Qualcomm delegates, plus packaged TorchAO kernels.

— Names delegates and kernels but no usage steps given v1.5.1

09 **Export pipeline, lowering and metadata improvements** IMPROVED

65

Adds multi-method model support to the export pipeline and per-backend load options, adds schema-version checks for PTE and PTD files, and folds redundant Q/DQ pairs in the shared EXIR `to_edge` lowering path.

— Names file formats and lowering path, no example v1.5.1

THINNER COVERAGE BELOW

10 **Windows ARM64 and CUDA wheel packaging** NEW

50

Adds Windows ARM64 build coverage and CUDA wheel publishing on pytorch.org.

— Names platforms, no install instructions v1.5.1

11 **Capacity-aware memory planning** NEW

50

Adds capacity-aware memory planning over target memory maps and support for shared allocations with offsets.

— Describes mechanism briefly, no config example v1.5.1

12 **BF16 support in XNNPACK LLM Export** NEW 45

Adds BF16 fully connected delegation and BF16 activation support in XNNPACK LLM Export.

— States feature but no mechanism or benchmark v1.5.1

13 **Public get_thread_count() runtime API** NEW 40

Adds the public get_thread_count() API to the runtime.

— Bare API name, no usage or context v1.5.1

v1.5.1 NOTES ↗ LATEST STABLE

ExecuTorch v1.5.1 adds a C++ SDK, ROCm backend, public get_thread_count() API, and broad new operator and model coverage.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.5.1
https://github.com/pytorch/executorch.git
# already have the repo? check out this version:
$ git checkout v1.5.1
```

Adds the public get_thread_count() API to the runtime.

Adds a C++ SDK to the Python wheel with linkable libraries for CUDA, Core ML, MLX, OpenVINO, and Qualcomm delegates and packaged TorchAO kernels.

Adds opt-in ROCm support to the AOTI CUDA backend, including packed W4 execution and Voxtral Realtime coverage on gfx1100 and gfx950.

Adds multi-method model support to the export pipeline and per-backend load options.

Adds schema-version checks for PTE and PTD files.

+23 more

WAS THIS USEFUL? 👍 👎

Ollama

Feedback ⓘ v0.34.4 NOTES ↗ CODE ↗ RANK

Local large language model runner that downloads, serves, and chats with models through a CLI and REST API.

> Ollama v0.34.4 makes structured outputs on thinking models faster and more reliable, and improves Gemma 4 image handling on Apple Silicon.

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

01 **Adaptive image resolution for Gemma 4 on Apple Silicon**

IMPROVED 65

Gemma 4 on Apple Silicon now picks the best image resolution per image across the 70, 140, 280, 560, and 1120 token budgets, preserving more detail in high-resolution images while using fewer tokens for smaller ones.

– Names concrete token budgets but no config surface to invoke v0.34.4

THINNER COVERAGE BELOW

02 **Single-pass structured outputs on thinking models** IMPROVED 35

Structured outputs on thinking models now apply in a single pass, making them faster and more reliable than the previous approach.

– Explains the mechanism change but no flags or numbers v0.34.4

v0.34.4 NOTES ↗ CODE ↗ LATEST STABLE

Ollama v0.34.4 speeds up structured outputs on thinking models and improves Gemma 4 and Qwen 3.8 performance on Apple Silicon.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.34.4 https://github.com/ollama/ollama.git
# already have the repo? check out this version:
$ git checkout v0.34.4
```

Structured outputs on thinking models now apply in a single pass, making them faster and more reliable.

Gemma 4 on Apple Silicon now picks the best image resolution per image across the 70, 140, 280, 560, and 1120 token budgets, preserving more detail in high-resolution images while using fewer tokens for smaller ones.

WAS THIS USEFUL? 👍 👎

Together AI

Feedback ⓘ changelog · Sep 24, 2026 CHANGELOG ↗

Run, train, and serve open-source AI models on Together AI.

Together AI: Adds `tg beta endpoints rollout` CLI command and `client.beta.endpoints.rollouts` Python SDK surface to shift live traffic between deployments under the same endpoint URL using canary, blue-green, or rolling strategies, with optional metric-gated pause.

Adds rollout controls (pause, resume, promote, cancel) accessible from the endpoint's Rollouts tab in the console.

└──▷ **BREAKING ON UPGRADE**

! The management API now rejects unknown JSON request-body fields and unknown query parameters with an HTTP error (naming the field, e.g. `unknown field "inactive_timeout"`); clients that send retired or misspelled keys will receive errors instead of silent success.

WAS THIS USEFUL? 👍 👎

Hugging Face

Feedback ⓘ

Hugging Face is now on Google Cloud Marketplace · NOTES ↗ · RANK

Hugging Face is an open platform for building, sharing, and deploying machine-learning models, datasets, and applications.

> Hugging Face added Google Cloud Marketplace billing for organizations and shipped synchronized multi-camera episode previews for LeRobot datasets.

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

01 **Google Cloud Marketplace billing integration** NEW 72

Hugging Face organizations can now subscribe via Google Cloud Marketplace to consolidate usage — Inference Providers, Inference Endpoints, Spaces, Jobs, storage, and Team or Enterprise seat costs — onto an existing GCP invoice instead of a separate vendor or credit card.

Consolidate Hugging Face spend onto your existing Google Cloud invoice — no separate vendor or credit card required.

📍 1. Go to the Hugging Face listing on Google Cloud Marketplace. 2. Click Subscribe and authenticate with your Google Cloud account. 3. Select the Hugging Face organization to link to your GCP billing account. 4. Wait for subscription validation — all usage (Inference Endpoints, Spaces, Jobs, storage, seats) will then appear on your GCP invoice.

— Names billed surfaces and gives exact subscription steps.

Hugging Face is now on Google Cloud Mar…

THINNER COVERAGE BELOW

02 **Synchronized multi-camera previews for LeRobot datasets** NEW 58

LeRobot dataset pages now open with a synchronized multi-camera episode preview that lets users scrub through an episode and see all camera angles at the same moment. The episode list previews each take on hover, and dataset listing cards for LeRobot datasets now show robot type, episode count, and a preview of the first episode inline.

Browse all LeRobot datasets and use the inline episode preview to evaluate robot type and recording content before opening a dataset.

📍 In the Hugging Face dataset hub, go to `https://huggingface.co/datasets?other=LeRobot` — hover over any dataset card to see its first episode preview, robot type, and episode count, then

open a dataset page to scrub through episodes across all camera angles simultaneously.

— Describes mechanism and UI path but no config or API surface. Preview LeRobot Episodes

Hugging Face is now on Google Cloud Marketplace NOTES ↗ STABLE

Hugging Face now available on Google Cloud Marketplace, consolidating HF usage onto existing GCP billing.

└──▷ **HOW TO FIND IT**

Consolidate Hugging Face spend onto your existing Google Cloud invoice — no separate vendor or credit card required.

📍 1. Go to the Hugging Face listing on Google Cloud Marketplace. 2. Click Subscribe and authenticate with your Google Cloud account. 3. Select the Hugging Face organization to link to your GCP billing account. 4. Wait for subscription validation — all usage (Inference Endpoints, Spaces, Jobs, storage, seats) will then appear on your GCP invoice.

Preview LeRobot Episodes NOTES ↗ LATEST STABLE

LeRobot dataset pages now show synchronized multi-camera episode previews with hover-to-preview episode lists.

└──▷ **HOW TO FIND IT**

Browse all LeRobot datasets and use the inline episode preview to evaluate robot type and recording content before opening a dataset.

📍 In the Hugging Face dataset hub, go to `https://huggingface.co/datasets?other=LeRobot` — hover over any dataset card to see its first episode preview, robot type, and episode count, then open a dataset page to scrub through episodes across all camera angles simultaneously.

LeRobot dataset pages now open with a synchronized multi-camera episode preview, letting you scrub through an episode and see all camera angles at the same moment.

The episode list on a LeRobot dataset page previews each take on hover, so you can scan recordings without opening them individually.

Dataset listing cards for LeRobot datasets now surface robot type, episode count, and a preview of the first episode inline.

WAS THIS USEFUL?  

◆ Local LLM Runtimes

llama.cpp

Feedback ⓘ v0.5.0 NOTES ↗ CODE ↗ RANK

Llama CPP is a C/C++ library and set of command-line tools that runs large language models locally.

> llama.cpp v0.5.0 adds multi-address HTTP server binding, a LoRA-from-file API, and multimodal tool-call output support, alongside broad additions to model architecture/conversion coverage, chat parsers, and backend performance optimizations.

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

01 Multi-address HTTP binding for llama-server NEW

80

The `--host` flag now accepts comma-separated TCP addresses and UNIX sockets, letting a single llama-server instance bind to multiple network interfaces simultaneously (e.g. a LAN address and a local UNIX socket at once).

Bind a single llama-server instance to both a LAN interface and a local UNIX socket simultaneously, so internal tooling and external clients share one model process.

```
$ llama serve -hf ggml-org/Qwen3.5-0.8B-GGUF --host
192.168.1.10,127.0.0.1,/tmp/llama.sock
```

— Named flag with a runnable multi-target command example. v0.5.0

02 New model architecture and conversion support NEW

65

Adds `--fuse-qkv` conversion flag support for the Muse Glimmer architecture, HRM-Text / DFM Mimir 1B model architecture support, MiMo-V2.6 model conversion support, DFlash support for HunyuanOCR model conversion, extended Nemotron MTP and Nemotron-H model handling, and Qwen4Exp hyper-connection ops with sparse flash attention support.

— Many named architectures/flags but no usage steps. v0.5.0

THINNER COVERAGE BELOW

03 LoRA adapter loading from open file pointer NEW

55

Adds the llama_adapter_lora_init_from_file_ptr() C API function for loading LoRA adapters directly from an already-open `FILE` pointer.

— Named API function but no usage example given. v0.5.0

04 **Backend performance optimizations** IMPROVED 55

Accelerates CUDA `conv2d` via implicit GEMM for vision/convolutional workloads, adds Metal MoE and `SSM_CONV` fusion optimizations for Apple Silicon, and enables CUDA graphs for MTP (multi-token prediction) drafting.

— Names optimizations and backends but no benchmarks. v0.5.0

05 **Environment-variable sampler controls** NEW 50

Adds environment variables for the `temperature`, `top-p`, `min-p`, and penalties sampler parameters, allowing inference defaults to be set without CLI flags.

— Named parameters but exact env var names not given. v0.5.0

06 **Chat parser support for new model formats** NEW 50

Adds chat parser support for Ling 3.0, DeepSeek V3.2/V4, qwen3-coder, Muse Glimmer, and Gemma 4.

— Lists five named formats but no mechanism. v0.5.0

07 **ggml core library updated to v0.25.0** IMPROVED 50

Updates ggml to v0.25.0, adding hyper-connection, flash-attention, and fused MoE/SSM support across backends, plus RPC protocol major version `v7`.

— Names version and protocol bump but no migration steps. v0.5.0

08 **input_image in function-call outputs** NEW 40

Adds `input_image` support to server function-call outputs, enabling multimodal tool-call responses that include image data.

— Names the field but no mechanism or example. v0.5.0

09 **LLAMA_VOCAB_TYPE_TEST dummy tokenizer** NEW 40

Adds the `LLAMA_VOCAB_TYPE_TEST` dummy tokenizer for use in testing and tooling pipelines.

— Named constant, minimal detail on use. v0.5.0

10 **Scheduler graph allocation diagnostics** NEW 40

Adds graph input and input-tensor diagnostics during scheduler reserve for debugging compute graph allocation.

— Thin description, no usage path given. v0.5.0

11 **llama-bench --version flag** NEW 40

Adds a `--version` flag to the `llama-bench` benchmark tool.

— Named, runnable flag but minimal context. v0.5.0

12 **WEBM video input in server UI** NEW 35

The server UI now accepts WEBM video files for multimodal input.

— Names format and surface, minimal further detail. v0.5.0

13 **ufakzeka pre-tokenizer support** NEW 25

Adds ufakzeka pre-tokenizer support.

— Bare mention with no further detail. v0.5.0

v0.5.0 NOTES ↗ CODE ↗ LATEST STABLE

llama.cpp v0.5.0 adds multi-address HTTP binding, `input_image` in function-call outputs, LoRA-from-FILE API, and env-var sampler controls.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.5.0 https://github.com/ggml-
org/llama.cpp.git
# already have the repo? check out this version:
$ git checkout v0.5.0
```

└──▷ TRY IT

Bind a single llama-server instance to both a LAN interface and a local UNIX socket simultaneously, so internal tooling and external clients share one model process.

```
$ llama serve -hf ggml-org/Qwen3.5-0.8B-GGUF --host
192.168.1.10,127.0.0.1,/tmp/llama.sock
```

Adds `--host` support for comma-separated TCP addresses and UNIX sockets, enabling a single server instance to bind to multiple network interfaces simultaneously.

Adds llama_adapter_lora_init_from_file_ptr() C API function for loading LoRA adapters directly from an already-open `FILE` pointer.

Adds environment variables for `temperature`, `top-p`, `min-p`, and penalties sampler parameters, allowing inference defaults to be set without CLI flags.

Adds `input_image` support to server function-call outputs, enabling multimodal tool-call responses that include image data.

Adds `LLAMA_VOCAB_TYPE_TEST` dummy tokenizer for use in testing and tooling pipelines.

+15 more

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ changelog · Sep 24, 2026

CHANGELOG ↗ RANK

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

> LangSmith shipped a `langsmith apps` CLI for building custom in-platform UIs, a shared read-only sandbox snapshot catalog, a new `/v2/threads/stats` API for self-hosted deployments, and tightened permissions on dataset exports, alongside smaller additions to IAM-based ECR access, license-expiration warnings, OAuth session controls, and LLM Gateway provider tagging.

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

01 **`langsmith apps` CLI for custom UIs** NEW 85

New `langsmith apps` CLI commands let teams scaffold, run locally, and publish custom apps — UIs that run inside LangSmith and call the LangSmith API. Typical flow: `langsmith apps scaffold my-app`, `cd my-app`, `langsmith apps run`, `langsmith apps publish`.

Scaffold a new custom app, develop it locally, then publish it into LangSmith so your team can access the UI from within the platform.

```
$ langsmith apps scaffold my-app
cd my-app
langsmith apps run
langsmith apps publish
```

— Full CLI workflow with exact commands provided. product docs

THINNER COVERAGE BELOW

02 **Permission check on dataset exports** BREAKING 55

Dataset CSV and JSONL exports now enforce a download permission check; API callers that previously exported datasets with only read permission will now receive a forbidden response.

— Clear before/after but no endpoint or migration path named. snapshot-20260924

03 **IAM role-based ECR access for self-hosted** NEW 30

Self-hosted LangSmith deployments can now use IAM role-based access to ECR.

— Named only in summary, no mechanism or config given. snapshot-20260924

04 **Scanning projects with special-character names** IMPROVED

30

The engine now scans projects whose names contain punctuation, Unicode, or other special characters.

— Single-line description with no further detail. changelog-20260924-d394ac7c

05 **Expanded OAuth session controls** IMPROVED

10

LangSmith adds expanded OAuth session controls.

— Mentioned only in summary with no specifics. snapshot-20260924

└──▷ **BREAKING ON UPGRADE**

! Dataset CSV and JSONL exports now enforce a download permission check; API callers that previously exported datasets with only read permission will receive a forbidden response.

changelog · Sep 24, 2026 CHANGELOG ↗ LATEST STABLE

LangSmith: Create sandboxes from a shared, read-only catalog of system snapshots: `system/default:latest`, `system/custom-apps:latest` (with Smith Apps tooling), and `system/code-evaluators:latest` (with NumPy, pandas, jsonschema, SciPy, and scikit-learn preinstalled).

Engine now scans projects whose names contain punctuation, Unicode, or other special characters.

snapshot-20260924 NOTES ↗ STABLE

LangSmith adds `/v2` /threads/stats endpoint, IAM role-based ECR access, `LICENSE_EXPIRATION_WARNING_DAYS` config, and expanded OAuth session controls.

└──▷ **TRY IT**

Pull aggregated latency, token, and cost statistics for a tracing project to feed into a cost-monitoring dashboard.

```
$ curl -X GET 'https://<langsmith-host>/v2/threads/stats?project_id=<project_id>' \
  -H 'Authorization: Bearer <api_key>'
```

Set a 14-day lead time on the license expiration banner so self-hosted teams have more runway to renew before the warning appears.

.env

```
LICENSE_EXPIRATION_WARNING_DAYS=14
```

Tag a custom Bedrock model in the LLM Gateway so its traces carry provider metadata and map to provider-specific pricing.

```json
{
  "model": "my-bedrock-llama3",
  "base_url": "https://bedrock.us-east-1.amazonaws.com",
  "ls_provider": "bedrock"
}
```

Supports `GET /v2/threads/stats` on self-hosted deployments backed by ClickHouse, including when SmithDB queries are disabled.

└──▷ **BREAKING ON UPGRADE**

! Dataset CSV and JSONL exports now enforce a download permission check; API callers that previously exported datasets with only read permission will receive a forbidden response.

WAS THIS USEFUL? 👍 👎

Arize Phoenix

Feedback ⓘ arize-phoenix-v20.16.0 NOTES ↗

RANK

Arize Phoenix is an open-source platform that monitors LLM application traces and evaluates their outputs.

> Arize Phoenix's latest release adds chronological span sorting, new model cost entries for the playground, and two new JavaScript SDK helpers for dataset management and secrets.

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

01 **Span list sorted by start_time** IMPROVED 55

The span list endpoint now sorts results by `start_time`, enabling chronologically ordered span retrieval.

– Names exact field and endpoint behavior but no method/path arize-phoenix-v20.16.0

02 **New model cost entries for playground** NEW 50

Adds `gpt-6-sol`, `gpt-6-luna`, and `claude-opus-5-5` to the cost tables and the playground.

– Names specific models but no mechanism detail arize-phoenix-v20.16.0

03 **JS secrets management helper** NEW 15

Adds a JavaScript secrets management helper.

– Bare mention with no name, mechanism, or usage details arize-phoenix-v20.16.0

arize-phoenix-v20.16.0 NOTES ↗ LATEST STABLE

Adds JS dataset split write helpers, JS secrets management, span list sorting by start_time, and new model cost entries for gpt-6-sol, gpt-6-luna, and claude-opus-5-5.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch arize-phoenix-v20.16.0
https://github.com/Arize-ai/phoenix.git
# already have the repo? check out this version:
$ git checkout arize-phoenix-v20.16.0
```

Sorts the span list endpoint by `start_time`, enabling chronologically ordered span

retrieval.

Adds `gpt-6-sol`, `gpt-6-luna`, and `claude-opus-5-5` to the cost tables and the playground.

Adds JavaScript dataset split write helpers for programmatic dataset management.

Adds a JavaScript secrets management helper.

WAS THIS USEFUL? 👍 👎

Comet Opik

Feedback ⓘ 2.2.78 NOTES ↗ RANK

Opik traces, evaluates, and monitors LLM applications and agentic workflows, with datasets, an LLM-as-judge metric library, and production dashboards.

> Opik 2.2.78 speeds up experiment item retrieval by 33x and adds an alert trigger configuration UI with inline webhook testing.

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

01 Faster experiment item reads via pagination IMPROVED

65

Reading experiment items is now 33x faster, achieved through pagination of 2,000 items per page, concurrent fetching, and removal of model parsing overhead.

— Names mechanism and exact numbers, no user-facing command 2.2.78

THINNER COVERAGE BELOW

02 Alert trigger configuration with webhook testing NEW

50

The alerts UI now supports alert trigger configuration with suggested names and inline webhook testing.

— Describes UI feature but no exact navigation path given 2.2.78

2.2.78 NOTES ↗ LATEST STABLE

Opik 2.2.78 adds alert trigger UI with inline webhook testing and reads experiment items 33x faster.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch 2.2.78 https://github.com/comet-ml/opik.git
# already have the repo? check out this version:
$ git checkout 2.2.78
```

Reads experiment items 33x faster via pagination of 2,000 items per page with concurrent fetching and no model parsing overhead.

Adds alert trigger configuration with suggested names and inline webhook testing to the alerts UI.

WAS THIS USEFUL?  

agentacct

Feedback ⓘ v0.12.0 NOTES ↗ RANK

The agentacct dashboard records coding-agent work steps, tools, file changes, tests, time, and token costs locally.

> agentacct adds a full terminal dashboard (`agentacct tui`) and a `agentacct capabilities agents` coverage command, alongside new Work, Dashboard, Sessions, and Usage views, Reported vs Verified outcome tracking, and saved Work Receipt snapshots with background updates.

└──▷ **WHAT SHIPPED · 8 FEATURES** most completely described first ? what's the number?

01 `agentacct tui` terminal dashboard command NEW

85

Adds `agentacct tui` subcommand that launches a shell dashboard combining receipts, usage, capacity, and a review overview, including a folder-anchored Work tab with a cross-agent timeline navigable by keyboard (↵ to open a task, ? for key bindings).

Launch the TUI to monitor all agents across projects in real time, then open the Work tab to scrub a cross-agent timeline.

```
$ agentacct tui
```

Launch the terminal UI to review cross-agent session receipts, usage, and the Work tab timeline from your shell.

```
$ agentacct tui
```

— Names exact command and its key bindings product docs

02 `agentacct capabilities agents` coverage command NEW

85

Adds `agentacct capabilities agents` command that prints the per-capability coverage matrix for every supported agent (Claude Code, Codex, OpenCode, Hermes, DeepSeek Harness, Kimi Code, OpenClaw, Cursor) as reported by the installed version.

Check which receipt lanes (tool calls, checks, cost, etc.) your installed version actually supports for each agent before relying on them in a report.

```
$ agentacct capabilities agents
```

Check which capabilities and coverage lanes are available for each supported agent in your installed version.

```
$ agentacct capabilities agents
```

— Exact command plus full supported-agent list `product docs`

03 **Usage view token and cost analytics** IMPROVED 78

The Usage view shows daily token charts broken down by client and model, separating cache reads and writes, with provider-reported quota windows and reset times on the same page. Selecting a chart bar or date opens a per-day drilldown with Fresh, Cache read, Cache write, Total, and Cost columns, a Tokens/Cost chart toggle that preserves the selected date when switching, and a Fresh/All token-chart preference where `All tokens` counts cache reads, cache writes, fresh input, and fresh output once each and is remembered across sessions.

— Detailed mechanism but reachable only via UI `v0.12.0`

04 **Work groups for project-scoped agent activity** IMPROVED 69

Work groups point at a project folder to show which agents worked there, session details, a shared cross-agent timeline, and group token/cost totals, with partial coverage and overlap labeled. The Work view also supports searching a project's sessions without first creating a group, and an opened group shows client participation, recent sessions, recorded tool calls and checks, tokens, cost, and the shared timeline.

— Describes group mechanics but only UI navigation `v0.12.0`

05 **Sessions view with Work Receipts and search** NEW 69

Adds a Sessions view with a latest-first task list and search by task or project, plus per-task Work Receipts combining participating sessions, recorded outcome, estimated cost, claim coverage, check results, and a full activity timeline. Sessions defaults to Latest, and adds `Recorded issues` and `With issues` sub-views so historical failures remain available without taking over the default view.

— Detailed view mechanics, UI navigation only `v0.12.0`

06 **Dashboard overview of recent agent activity** IMPROVED 61

The Dashboard shows the latest recorded work across projects with client, activity time, outcome, evidence, and cost, and includes a Recorded issues panel (right-click to copy a

brief from recorded facts). Recent project and client activity is now surfaced first in the Dashboard, Work, and Sessions views.

— Names panel interaction but only via UI v0.12.0

07 **Reported vs Verified outcome tracking** NEW 60

Introduces Reported vs. Verified outcome states: Reported reflects the agent's own `done` claim, while Verified requires current passing checks after the last recorded change; failed attempts are preserved in history for context.

— Mechanism explained, no command or key named product docs

THINNER COVERAGE BELOW

08 **Saved Work Receipt snapshots with background updates** IMPROVED 55

Work Receipt snapshots are now saved so opening a receipt no longer waits for a full rebuild; a background worker prepares updates while the last safe persisted snapshot is shown immediately, along with its capture time and preparation status.

— Explains mechanism, no command or config named v0.12.0

v0.12.0 NOTES ↗ LATEST STABLE

agentacct v0.12.0 adds date-drilldown usage breakdowns, a Fresh/All token toggle, and saved Work Receipt snapshots with background updates.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.12.0
https://github.com/mikehasa/agentacct.git
# already have the repo? check out this version:
$ git checkout v0.12.0
```

Adds per-day usage drilldown in the Usage view: select a chart bar or date to see client and model breakdown with Fresh, Cache read, Cache write, Total, and Cost columns.

Adds a Tokens/Cost chart toggle in the Usage view, with the selected date preserved when switching.

Adds a Fresh/All token-chart preference in Usage, where `All tokens` includes cache reads, cache writes, fresh input, and fresh output counted once each; the preference is remembered across sessions.

Saves Work Receipt snapshots so opening a receipt no longer waits for a full rebuild; a background worker prepares updates while the last safe persisted snapshot is

> shown immediately, along with its capture time and preparation status.
>
> Surfaces recent project and client activity first in the Dashboard, Work, and Sessions views, with Sessions defaulting to Latest and supporting project search.
>
> +2 more

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ changelog · Sep 24, 2026 CHANGELOG ↗

RANK

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

> Braintrust's biggest window addition is automated trace analysis (Pattern frequency charts and the Debugger) alongside project-wide Loop automations and thread sharing, plus TypeSafe Jev as a structured-decision judge, columnar metric storage, an async Brainstore log search engine, Node.js 24 sandbox support, bt CLI tracing for Grok and Google Antigravity sessions, DeepSeek V4.1 Flash, RubyLLM 2.0 support, and new S3 attachment-uploader tuning controls.

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

01 **Debugger for single-trace failure analysis** NEW 70

Adds the Debugger, which reports likely failure modes for a single trace citing relevant spans, tool calls, and model outputs, alongside an Analyze trace view that groups a run into Work sections; requires data plane v2.12.0+.

— Names UI elements and data-plane requirement. snapshot-20260924

02 **Async log search engine in Brainstore** IMPROVED 70

Improves log search with an asynchronous execution engine in Brainstore that streams results as segments finish scanning, covering full-text, phrase, and multi-clause ANY_SPAN() trace searches including negated clauses; requires data plane v2.13.0+.

— Names engine and search syntax, no direct command. snapshot-20260924

03 **bt CLI tracing for Grok and Google Antigravity sessions** NEW 65

Adds `bt` CLI v0.19.3+ support for tracing Grok and Google Antigravity coding sessions as Braintrust traces, capturing prompts, model calls, and tool activity without interrupting the agent if tracing fails.

— Names CLI version and coding agents traced. snapshot-20260924

THINNER COVERAGE BELOW

04 **Context section in span detail panel** NEW 55

Adds a Context section to the span detail panel for spans that include a `context` field, such as origin data recorded by SDK integrations.

— Names field and UI location, no further mechanism. product docs

05 **Pattern frequency charts for trace findings** NEW 50

Adds pattern frequency charts showing how often a finding occurs among comparable traces across a selected time range, refreshed by scheduled runs; requires data plane v2.15.0+.

— Explains mechanism but no UI navigation given. snapshot-20260924

06 **Loop automations for scheduled runs** NEW 50

Adds Loop automations for scheduled, recurring Loop runs with their own instruction, model, and write permissions, without requiring a user to start a chat.

— Describes mechanism but no config surface named. snapshot-20260924

07 **Loop thread sharing for team investigations** NEW 50

Adds Loop thread sharing so project members can view or continue an investigation together, with view-only or chat access; requires data plane v2.15.0+.

— Names access levels, no exact UI path. snapshot-20260924

08 **Node.js 24 sandbox for code functions and scorers** NEW 40

Supports pushing and running code functions and scorers from Node.js 24, executed in a matching `nodejs24.x` sandbox.

— Names sandbox version but no usage detail. changelog-20260924-17f0e119

09 **RubyLLM 2.0 support in SDK** NEW 15

Traces RubyLLM 2.0 calls automatically in the Ruby SDK, capturing model invocations and their metadata from the RubyLLM framework.

— Just names version, no detail on what changed. product docs

└──▷ **BREAKING ON UPGRADE**

! On Braintrust-hosted deployments, project providers now override organization providers only for models they serve; organization providers can serve other models using their own credentials and billing. If you rely on project providers to restrict model access or spending, review organization provider model selections and permissions.

! The `--relabel-overlap` flag is removed from `bt topics config enable` and `bt topics config set`; it was deprecated and ignored since data plane v2.15.0. See the v0.20.0 migration instructions.

! On Braintrust-hosted deployments, project providers now override organization providers only for models they explicitly serve — organization providers can serve other models using their own credentials and billing accounts. If you relied on project providers to restrict model access or spending across all models, review organization provider model selections and permissions.

changelog · Sep 24, 2026 CHANGELOG ↗ LATEST STABLE

Braintrust: Adds `BRAINTRUST_ATTACHMENT_UPLOADER_QUEUE_SIZE`, `BRAINTRUST_ATTACHMENT_UPLOADER_REQUEST_TIMEOUT_MILLIS`, `BRAINTRUST_ATTACHMENT_UPLOADER_MAX_RETRIES`, and `BRAINTRUST_ATTACHMENT_UPLOADER_INITIAL_RETRY_DELAY_MILLIS` environment variables (via `BraintrustConfig`) to tune S3 attachment uploader behavior including queue size, request timeout, retry count, and initial retry delay.

└──▷ **TRY IT**

Set S3 attachment uploader retry behavior via environment variables to avoid dropped attachments under transient network failures.

```
$ export BRAINTRUST_ATTACHMENT_UPLOADER_MAX_RETRIES=5
export BRAINTRUST_ATTACHMENT_UPLOADER_REQUEST_TIMEOUT_MILLIS=10000
export BRAINTRUST_ATTACHMENT_UPLOADER_INITIAL_RETRY_DELAY_MILLIS=500
export BRAINTRUST_ATTACHMENT_UPLOADER_QUEUE_SIZE=200
```

Adds `nodejs24.x` sandbox support for pushing and running code functions and scorers from Node.js 24.

Adds TypeSafe Jev as a judge in scorers and classifiers, supporting built-in models or a user-supplied TypeSafe key, recording confidence and per-choice probabilities for each decision.

└──▷ **BREAKING ON UPGRADE**

! On Braintrust-hosted deployments, project providers now override organization providers only for models they serve; organization providers can serve other models using their own credentials and billing. If you rely on project providers to restrict model access or spending, review organization provider model selections and permissions.

! The `--relabel-overlap` flag is removed from `bt topics config enable` and `bt topics config set`; it was deprecated and ignored since data plane v2.15.0. See the v0.20.0 migration instructions.

snapshot-20260924 NOTES ↗ STABLE

Braintrust adds automated trace analysis (Patterns, Debugger), Loop project-wide automation, Jev structured decisions, columnar metric storage, and DeepSeek V4.1 Flash.

└──▷ **HOW TO FIND IT**

Enable columnar metric storage so Logs page queries scan far less data on a high-volume project.

📍 In the Braintrust UI, go to your project Settings › Metric columnstore and enable it alongside log search optimization.

Adds `bt` CLI v0.19.3+ support for tracing Grok and Google Antigravity coding sessions as Braintrust traces, capturing prompts, model calls, and tool activity without interrupting the agent if tracing fails.

Adds pattern frequency charts showing how often a finding occurs among comparable traces across a selected time range, refreshed by scheduled runs; requires data plane v2.15.0+.

Adds the Debugger, which reports likely failure modes for a single trace citing relevant spans, tool calls, and model outputs, alongside an Analyze trace view that groups a run into Work sections; requires data plane v2.12.0+.

Adds Loop automations for scheduled, recurring Loop runs with their own instruction, model, and write permissions, without requiring a user to start a chat.

Adds Loop thread sharing so project members can view or continue an investigation together, with view-only or chat access; requires data plane v2.15.0+.

+1 more

└─▷ **BREAKING ON UPGRADE**

! On Braintrust-hosted deployments, project providers now override organization providers only for models they explicitly serve — organization providers can serve other models using their own credentials and billing accounts. If you relied on project providers to restrict model access or spending across all models, review organization provider model selections and permissions.

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

AI-Infra-Guard

Feedback ⓘ v4.6.3 NOTES ↗ RANK

AI-Infra-Guard is an AI red teaming platform that scans AI infrastructure for CVE vulnerabilities, MCP/Agent skill risks, jailbreak weaknesses, and API relay security issues.

> AI-Infra-Guard's v4.6.3 release focused on reliability fixes: DeepTeam now persists test cases mid-run, and API Checker's model fingerprinting was made more robust.

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

01 **Mid-run disk persistence for DeepTeam test cases** IMPROVED

40

DeepTeam now writes finished test cases to disk during a run instead of only at completion, adding mid-run disk persistence so progress is not lost if a run is interrupted before finishing.

— Explains behaviour change but no config/flag named v4.6.3

02 **API Checker model fingerprinting reliability** IMPROVED

28

API Checker improves model identification reliability by refilling failed fingerprint samples.

— Mechanism named briefly, no concrete usage path v4.6.3

v4.6.3 NOTES ↗ LATEST STABLE

DeepTeam now writes test cases to disk mid-run; API Checker gets more reliable model fingerprinting.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.6.3 https://github.com/Tencent/AI-Infra-Guard.git
# already have the repo? check out this version:
$ git checkout v4.6.3
```

Adds mid-run disk persistence for DeepTeam test cases, writing finished cases to disk during a run instead of only at completion.

Improves API Checker model identification reliability by refilling failed fingerprint samples.

WAS THIS USEFUL?  

NVIDIA SkillSpector

Feedback ⓘ v2.12.0 NOTES ↗ RANK

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

> SkillSpector v2.12.0 adds a CI-friendly hard-fail flag, deeper dependency-source and Python-reflection detection, an OpenCode integration for semantic analysis, and two breaking changes to how severity and reference findings are reported.

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

01 **`--fail-on-findings` CI gate** NEW 80

Adds `skillspector scan --fail-on-findings` to exit with code 1 when any active finding is present, including findings below the default risk-score threshold, across single-skill, recursive, and MCP registry scans.

Fail CI immediately when a scan of a PR's skill directory surfaces any active finding, regardless of risk-score threshold.

```
$ skillspector scan ./my-skill/ --fail-on-findings
```

— Exact flag and command given, scope across scan types explained v2.12.0

02 **OpenCode integration for semantic analysis** NEW 80

Adds `SKILLSPECTOR_PROVIDER=opencode_cli` to run semantic analysis through a local OpenCode login, requiring exactly OpenCode 1.18.31 and failing closed on auth, version, policy, or empty-output failures. Also adds a native `/skillspector` command and `skillspector_scan` tool via a repository-provided OpenCode extension, installed by copying `.opencode/` from a checkout, which resolves the binary from `SKILLSPECTOR_BIN`, a worktree `.venv`, or `PATH`.

— Names env var, version requirement, extension files, install step v2.12.0

03 **GitHub `/tree/` subdirectory scanning** NEW 75

Adds support for HTTPS GitHub `/tree/` URLs to select an existing repository subdirectory as scan input, resolving using the longest advertised branch/tag name including refs containing slashes.

Scan only the `src/` subdirectory of a GitHub repository by passing a `/tree/` URL directly.

```
$ skillspector scan https://github.com/example-org/my-skill/tree/main/src
```

— Exact command example plus resolution mechanism explained v2.12.0

04 **SC10 dependency-source detection** NEW 70

Adds SC10 detection of noncanonical or unresolved dependency sources in npm, Yarn, pip, Poetry, Maven, and Cargo configuration, supported shell scripts, generated heredocs, and actionable shell fences in `SKILL.md` / `README.md`.

— Names rule ID and every covered ecosystem, but no runnable example v2.12.0

05 **Configurable static-analysis time budget** IMPROVED 65

Adds `SKILLSPECTOR_MAX_STATIC_ANALYSIS_SECONDS_PER_ARTIFACT` to configure the per-artifact static-pattern and YARA time allowance; the default is raised from 30 to 300 seconds.

— Named env var and default change, but no usage example v2.12.0

06 **LLM provenance metadata in JSON reports** NEW 65

Adds `metadata.llm_provenance` (schema version 1) to JSON reports, recording sanitized, bounded model/package/source identity and distinguishing configured, resolved, and effective providers without endpoint URLs or credentials. Adds `SKILLSPECTOR_BUILD_REVISION` to supply source revision metadata for this field when unavailable from the installed package.

— Named field, schema version, and env var, but no example output v2.12.0

07 **Split `reference_missing`/`reference_unresolved` codes** 60

BREAKING

Distinguishes `reference_missing` (missing local references) from `reference_unresolved` (ambiguous matches) in MCP skill-scan installation decisions. This replaces what was previously reported as `reference_unresolved` for missing local file references, so tooling matching on the `reference_unresolved` string must be updated to handle both values.

— Names both codes and migration impact clearly v2.12.0

THINNER COVERAGE BELOW

08 **GHSA/OSV MODERATE mapped to MEDIUM** BREAKING 55

Maps GHSA/OSV `MODERATE` severity to `MEDIUM` instead of the previous `LOW`, potentially increasing affected supply-chain risk scores and changing scan outcomes

that previously passed a `MEDIUM` `-or-higher` threshold.

– Names exact severity values changed and downstream threshold effect v2.12.0

09 **Compact LLM prompt line labels** NEW 50

Adds `SKILLSPECTOR_COMPACT_PROMPTS=1` (also accepting `true` or `yes`) to enable unpadded LLM prompt line labels such as `L1:` instead of `L01:`.

– Names env var and value format but limited context on impact v2.12.0

10 **JSON and SARIF column reporting changes** IMPROVED 50

JSON findings now retain each occurrence's own zero-based columns; SARIF emits one-based Unicode-codepoint columns for both active and suppressed findings.

– Precise format details but no example output shown v2.12.0

11 **Gemini 3.5 Flash model support** NEW 45

Adds `gemini-3.5-flash` model budgets to the root and OpenAI-compatible registries, usable through the existing `openai` provider.

– Names model and provider but no usage steps v2.12.0

12 **TP4 analysis of Markdown/text fences** IMPROVED 45

Extends TP4 semantic analysis to non-empty, closed Markdown and text fences with recognized executable-language labels.

– States scope change but no mechanism detail or example v2.12.0

13 **Live scan progress streaming** NEW 45

Streams bounded live scan progress, completed analyzer rules, and a discovered-file tree to standard error during non-verbose interactive terminal scans.

– Describes behaviour but no flag or config to control it v2.12.0

14 **Reflective Python module and bytecode detection** NEW 35

Adds detection of reflective Python module lookups and shipped bytecode inside Python environments.

– Bare capability description, no mechanism or usage detail v2.12.0

└──▷ **BREAKING ON UPGRADE**

! `GHSA/OSV MODERATE` severity now maps to `MEDIUM` instead of `LOW`, which may increase risk scores for skills with supply-chain findings and change scan outcomes that previously passed a `MEDIUM` `-or-higher` threshold.

! The `reference_missing` code replaces what was previously reported as `reference_unresolved` for missing local file references; tooling that matches on the

`reference_unresolved` string must be updated to handle both values.

v2.12.0 NOTES ↗ LATEST STABLE

SkillSpector v2.12.0 adds `--fail-on-findings` gate, SC10 dependency-source detection, OpenCode integration, Gemini 3.5 Flash support, and TP4 Markdown fence analysis.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v2.12.0
https://github.com/NVIDIA/SkillSpector.git
# already have the repo? check out this version:
$ git checkout v2.12.0
```

└──▷ **TRY IT**

Fail CI immediately when a scan of a PR's skill directory surfaces any active finding, regardless of risk-score threshold.

```
$ skillspector scan ./my-skill/ --fail-on-findings
```

Scan only the `src/` subdirectory of a GitHub repository by passing a `/tree/` URL directly.

```
$ skillspector scan https://github.com/example-org/my-
skill/tree/main/src
```

Adds `skillspector scan --fail-on-findings` to exit with code 1 when any active finding is present, including findings below the default risk-score threshold, across single-skill, recursive, and MCP registry scans.

Adds `SKILLSPECTOR_MAX_STATIC_ANALYSIS_SECONDS_PER_ARTIFACT` to configure the per-artifact static-pattern and YARA time allowance; default raised from 30 to 300 seconds.

Adds `SKILLSPECTOR_PROVIDER=opencode_cli` to run semantic analysis through a local OpenCode login, requiring exactly OpenCode 1.18.31 and failing closed on auth, version, policy, or empty-output failures.

Adds `SKILLSPECTOR_COMPACT_PROMPTS=1` (also `true` or `yes`) to enable unpadded LLM prompt line labels such as `L1:` instead of `L01:`.

Adds `SKILLSPECTOR_BUILD_REVISION` environment variable to supply source revision metadata for `metadata.llm_provenance` when unavailable from the installed package.

+11 more

└─▷ **BREAKING ON UPGRADE**

! `GHSA/OSV MODERATE` severity now maps to `MEDIUM` instead of `LOW`, which may increase risk scores for skills with supply-chain findings and change scan outcomes that previously passed a `MEDIUM` `-or-higher` threshold.

! The `reference_missing` code replaces what was previously reported as `reference_unresolved` for missing local file references; tooling that matches on the `reference_unresolved` string must be updated to handle both values.

WAS THIS USEFUL?  

ai-safe2-framework

Feedback ⓘ

2026-09-23_NEXUS_v0.4-Safer_Payment_Authority_for_Agents NOTES ↗ RANK

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

> NEXUS v0.4 adds eight payment authority reference adapters spanning x402, AP2, Visa TAP, Mastercard AP4M, and KYA-OS, along with a three-state Supported/Contradicted/Unverifiable assessment model and an adversarial test corpus for adapter conformance.

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

01 **Payment authority reference adapters for agent transactions**

NEW 80

Adds eight reference adapters: `SafePay Exact` for structural x402 v2 exact EVM/USDC evaluation against canonical NEXUS transactions; `SafePay Verified` for authenticated x402 facilitator or chain evidence with an explicit verifier boundary and replay protection; `SafePay Variable` for maximum-versus-actual charge enforcement and reservation release; `SafePay Escrow` for deposit, release, refund, and mediated-finality state handling; Mandate Bridge for AP2 v0.2 mandate and receipt evidence; Card Trust Bridge for Visa TAP request recognition with explicit limits on what payment authorization may be inferred; Agent Token Bridge for authenticated Mastercard AP4M verifier evidence; and Portable Delegation Bridge for KYA-OS identity and delegated-authority evidence.

— Names eight adapters and their exact evaluation scope, but no invocation example given.

2026-09-23_NEXUS_v0.4-Safer_Payment_Aut…

THINNER COVERAGE BELOW

02 **Three-state assessment results for authority decisions** NEW

55

Introduces three-state assessment results — Supported, Contradicted, and Unverifiable — giving agents stable machine-readable authority decisions in place of permission inferred from prose.

— States the three named outcomes but no usage mechanism shown.

2026-09-23_NEXUS_v0.4-Safer_Payment_Aut…

03 **Adversarial test corpus for adapter conformance** NEW

55

Adds an adversarial test corpus covering substitution attacks, malformed evidence, truthy-value bypasses, replay behavior, scope mapping, privacy checks, and RailGuard conformance.

— Lists test categories but no way to run or invoke the corpus.

2026-09-23_NEXUS_v0.4-Safer_Payment_Aut…

2026-09-23_NEXUS_v0.4-Safer_Payment_Authority_for_Agents NOTES ↗ LATEST

STABLE

NEXUS v0.4 adds eight payment authority adapter profiles for AI agents, covering x402, AP2, Visa TAP, Mastercard AP4M, and KYA-OS delegation evidence.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch 2026-09-23_NEXUS_v0.4-
Safer_Payment_Authority_for_Agents
https://github.com/CyberStrategyInstitute/ai-safe2-framework.git
# already have the repo? check out this version:
$ git checkout 2026-09-23_NEXUS_v0.4-
Safer_Payment_Authority_for_Agents
```

Adds `SafePay Exact` reference adapter for structural x402 v2 exact EVM/USDC evaluation against canonical NEXUS transactions.

Adds `SafePay Verified` reference adapter for authenticated x402 facilitator or chain evidence with an explicit verifier boundary and replay protection.

Adds `SafePay Variable` reference adapter for maximum-versus-actual charge enforcement and reservation release.

Adds `SafePay Escrow` reference adapter for deposit, release, refund, and mediated-finality state handling.

Adds `Mandate Bridge` reference adapter for AP2 v0.2 mandate and receipt evidence.

+5 more

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Frontier Models

Anthropic

Feedback ⓘ changelog · Sep 24, 2026 CHANGELOG ↗ RANK

Anthropic provides Claude AI models and APIs for building applications that generate, analyze, and automate work.

> Anthropic launched Claude Opus 5.5 with mandatory extended thinking, mid-conversation tool schema updates, and on-demand context compaction, alongside new logging to support stricter thinking-prefix enforcement and a platform-specific tool deprecation.

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

01 Thinking mismatch logging for prefix-check rollout NEW

75

On `claude-fable-5-1`, responses can now surface `thinking_mismatch_allowed` entries within `thinking` content, letting teams detect history edits in production traffic before enabling strict prefix-check enforcement.

Detect history edits in production traffic before enabling strict prefix-check enforcement — log `thinking_mismatch_allowed` entries to identify problematic sessions.

```
$ curl https://api.anthropic.com/v1/messages \
  -H 'content-type: application/json' \
  -d '{
    "model": "claude-fable-5-1",
    "messages": [...]
  }' | jq '[.content[] | select(.type == "thinking") | .entries[]
| select(.type == "thinking_mismatch_allowed")]'
```

— Names exact log field and filter command, purpose of enforcement unclear

changelog-20260924-3250cc0a

02 Mandatory thinking on Claude Opus 5.5 BREAKING

70

On `claude-opus-5-5`, sending `thinking: {"type": "disabled"}` or omitting thinking config entirely now returns a 400 error — thinking can no longer be disabled on this model.

— Exact error condition given, no migration guidance beyond that

changelog-20260924-3250cc0a

└──▷ BREAKING ON UPGRADE

! On `claude-opus-5-5`, sending `thinking: {"type": "disabled"}` or omitting thinking config returns a 400 error — thinking cannot be disabled on this model.

! On `claude-opus-5-5` via the Claude API and Google Cloud, the earlier tool (superseded by the newer toolset) returns a 400 error; Amazon Bedrock continues to accept it.

changelog · Sep 24, 2026 CHANGELOG ↗ LATEST STABLE

Claude Opus 5.5 launches with mandatory thinking, inline tool definitions via mid-conversation system messages, and on-demand conversation compaction.

└──▷ **TRY IT**

Add or update a tool mid-conversation without invalidating your prompt cache — useful when a server-side tool schema changes between turns.

```
$ curl https://api.anthropic.com/v1/messages \
  -H 'anthropic-beta: inline-tools-2026-09-15' \
  -H 'content-type: application/json' \
  -d '{
    "model": "claude-opus-5-5",
    "messages": [
      {"role": "user", "content": "What is the weather in Paris?"},
      {
        "role": "system",
        "content": [
          {
            "type": "tool_addition",
            "tool": {
              "type": "tool_definition",
              "definition": {
                "name": "get_weather",
                "description": "Returns current weather for a
city.",
                "input_schema": {"type": "object", "properties":
{"city": {"type": "string"}}, "required": ["city"]}
              }
            }
          }
        ]
      }
    ]
  }'
```

Compact a long agentic conversation on demand to stay within context limits while preserving the most recent turns verbatim.

```
$ curl https://api.anthropic.com/v1/messages \
  -H 'anthropic-beta: compact-2026-09-04' \
  -H 'content-type: application/json' \
  -d '{
    "model": "claude-opus-5-5",
    "compaction": true,
    "messages": [
      {"role": "user", "content": "Summarize our progress so far and
continue."},
      ...
    ]
  }'
```

Detect history edits in production traffic before enabling strict prefix-check enforcement — log `thinking_mismatch_allowed` entries to identify problematic sessions.

```
$ curl https://api.anthropic.com/v1/messages \
  -H 'content-type: application/json' \
  -d '{
    "model": "claude-fable-5-1",
    "messages": [...]
  }' | jq '[.content[] | select(.type == "thinking") | .entries[] |
select(.type == "thinking_mismatch_allowed")]'
```

└─▷ **BREAKING ON UPGRADE**

! On `claude-opus-5-5`, sending `thinking: {"type": "disabled"}` or omitting thinking config returns a 400 error — thinking cannot be disabled on this model.

! On `claude-opus-5-5` via the Claude API and Google Cloud, the earlier tool (superseded by the newer toolset) returns a 400 error; Amazon Bedrock continues to accept it.

WAS THIS USEFUL? 👍 👎

Google Gemini API

Feedback ⓘ September 22, 2026 NOTES ↗

RANK

Gemini API provides Gemini models for generating text, images, and other multimodal content in applications.

> Gemini API's biggest shipment this window is GA text-to-speech: two new TTS models with voice design and replication and a new voices endpoint, alongside richer transcription configuration options and pricing for a wave of new agentic and grounding tools.

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

01 **Transcription configuration: language, vocabulary, diarization, timestamps**

NEW 85

`TranscriptionConfig` gains `LanguageCodes` to hint the language(s) of audio, `CustomVocabulary` to bias transcription toward domain-specific or uncommon terms, `DiarizationMode` (under `VerbatimTranscriptionMode`) for speaker diarization, and `TimestampGranularities` (also under `VerbatimTranscriptionMode`) to control word-level timestamp granularity. A new `TranscriptionConfigModeEnumSmart` mode is selectable via `TranscriptionConfig.Mode` alongside the existing verbatim mode, and `WordInfo` annotation support on `TextContent` exposes per-word speaker, `StartOffset`, and `EndOffset` fields in output.

— Every config field and enum named exactly, directly usable product docs

02 **Voices endpoint for the Extended Voice Library** NEW

75

Adds the `GET /v1beta/voices` Gemini API Voices endpoint to query 150+ prebuilt and custom voices from the Extended Voice Library.

— Exact method, path and voice count given September 22, 2026

03 **New pricing for tools, agents, and grounding services** NEW

75

Introduces pricing for a Custom Tools endpoint on Gemini 3.1 Pro Preview; File search billed for embeddings at $0.15/1M tokens with retrieved document tokens at standard per-model rates; Google Maps grounding with tiered pricing (1,500 RPD free for Flash/Flash-Lite, 10,000 RPD free for Pro, then $25 per 1,000 grounded prompts); Managed agents in Gemini API and Antigravity Agent as billable agent types with environment compute unbilled during preview; Gemini Deep Research agent billed at

standard Gemini list rates including intermediate reasoning tokens; agentic video understanding billing that samples tokens dynamically, cutting input tokens by up to 88% for long-form video; `DOCUMENT` modality tokens billed at the image token rate and surfaced under `DOCUMENT` in `promptTokensDetails`; and updated Google Search grounding pricing for Gemini 3 models (5,000 free requests/month shared across models, then $14 per 1,000 requests).

— Concrete rates and tiers for many surfaces, but informational not runnable `product docs`

04 **Gemini 3.8 Flash TTS models reach GA** NEW 70

Gemini API ships `gemini-3.8-flash-tts`, a flagship text-to-speech model for studio-grade voice fidelity, expressive acting, regional dialects, and long-form multi-turn stability, alongside `gemini-3.8-flash-lite-tts`, a fast, cost-efficient TTS model that replaces `gemini-3.1-flash-tts-preview` for high-throughput, low-latency production and real-time voice agent cascades.

— Names both model ids and the model they replace, but no code sample `September 22, 2026`

THINNER COVERAGE BELOW

05 **Voice Design and Voice Replication for TTS** NEW 45

Adds Voice Design to create persistent custom vocal personas from text prompts, and Voice Replication to clone voices with consent verification.

— Describes mechanism but no API surface to invoke it `September 22, 2026`

> `September 22, 2026` NOTES ↗ LATEST STABLE
>
> Gemini 3.8 Flash TTS and Flash-Lite TTS hit GA, adding voice design, replication, and a 150+ voice library via `/v1beta` /voices
>
> Adds the `GET /v1beta/voices` Gemini API Voices endpoint to query 150+ prebuilt and custom voices from the Extended Voice Library.
>
> Releases `gemini-3.8-flash-tts`, a flagship creative TTS model for studio-grade voice fidelity, nuanced acting, regional dialects, and long-form multi-turn stability.
>
> Releases `gemini-3.8-flash-lite-tts`, a fast, cost-efficient TTS model replacing `gemini-3.1-flash-tts-preview` for high-throughput production and real-time voice agent cascades.
>
> Adds Voice Design to create persistent custom vocal personas from text prompts.
>
> Adds Voice Replication to clone voices with consent verification.

WAS THIS USEFUL? 👍 👎

Mistral AI

Feedback ⓘ snapshot-20260924 NOTES ↗ RANK

Learn how to deploy and use Mistral AI's Large Language Models with our comprehensive documentation, guides, and tutorials.

> Mistral shipped a sweeping wave of model releases spanning flagship chat (Medium 3.5, Large 3, Small 4), coding (Codestral, Devstral), reasoning (Magistral), audio (Voxtral), and OCR (OCR 4.1 GA) models, alongside new Agents, Document Library, and Guardrails APIs plus a string of Completions and OCR API parameter additions.

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

01 **OCR API extraction and formatting parameters** IMPROVED

90

The OCR API gained a `"block"` value for `confidence_scores_granularity` (alongside `"page"` and `"word"`) returning block-level confidence scores; `include_blocks`, which when `true` returns a `blocks` array per page with paragraph-level bounding boxes and structural labels in reading order; `table_format` to choose `markdown` or `html` table output; `extract_footer` and `extract_header` parameters plus hyperlink output; and a `pages` parameter that now accepts comma-separated digit/range strings such as `"0,1,2"`, `"0-5"`, or `"0,2-4"` in addition to integer lists.

— Names every OCR parameter and value but no usage example snapshot-20260924

02 **Custom Guardrails for Agents and Conversations** NEW

81

Added Custom Guardrails support: configurable directly on an Agent via the `guardrails` parameter, or passed per-request on `POST /v1/conversations`.

— Names exact parameter and endpoint for direct use snapshot-20260924

03 **`p` parameter added to streamed completion chunks** BREAKING

78

Added parameter `p` to chunks streamed back by the Completion API; this may break applications relying on strict parsing of those chunks, so users of the `mistral-common` package must update to version 1.8.4 or higher.

— Names parameter, package and required version for migration snapshot-20260924

04 **Mistral Large, Medium and Small model family releases** NEW

73

Shipped Mistral Medium 3.5 (`mistral-medium-3-5`), a frontier-class multimodal model for agentic and coding use cases with an adjustable `reasoning_effort` parameter, released as open weights under a Modified MIT license; Mistral Small 4 (`mistral-small-2603`), a hybrid multimodal model unifying instruct, reasoning and coding with a 256k context window; Mistral Large 3 (`mistral-large-2512`) and the Ministral 3 family (`ministral-3b-2512` , `ministral-8b-2512` , `ministral-14b-2512`); Mistral Medium 3.1 (`mistral-medium-2508`) and Mistral Medium 3 (`mistral-medium-2505`); and Mistral Small 3.2 (`mistral-small-2506`), Mistral Small 3.1 (`mistral-small-2503`), and Mistral Small 3 (`mistral-small-2501`).

– Names each model id and key trait but no API usage shown `snapshot-20260924`

---

05 **Voxtral audio model family** NEW 73

Shipped the first audio chat model Voxtral Small (`voxtral-small-2507`) alongside a Transcription API, then Voxtral Mini (`voxtral-mini-2507`) for chat and Voxtral Mini Transcribe via `audio/transcriptions` ; later Voxtral Mini Transcribe 2 (`voxtral-mini-2602`) and Voxtral Mini Transcribe Realtime (`voxtral-mini-transcribe-realtime-2602`); and Voxtral TTS (`voxtral-tts-2603`), a text-to-speech model with zero-shot voice cloning, multilingual support and real-time streaming.

– Names each audio model id; TTS mechanism described, others bare `snapshot-20260924`

---

06 **Multiple completions per request via `n` parameter** NEW 69

Added an `n` parameter to the Completions API to return multiple completions per request, with input tokens billed only once.

– Names parameter and billing behavior, no code example `snapshot-20260924`

---

07 **OCR model releases and GA promotions** NEW 68

Mistral OCR 4.1 (`mistral-ocr-4-1`) is now Generally Available, with `mistral-ocr-latest` and `mistral-ocr-4` pointing to it; OCR 4 (`mistral-ocr-4-0`) also shipped with `mistral-ocr-4` pointing to it. Earlier releases added Mistral OCR 2 (`mistral-ocr-2505`) with annotations and the original Mistral OCR (`mistral-ocr-2503`) with document understanding.

– Names model ids and aliasing but no API mechanism detail `snapshot-20260924`

---

08 **Labs experimental model releases** NEW 65

Released Leanstral (`labs-leanstral-2603`), an open-source code agent for Lean 4 formal proof engineering, followed by Leanstral 1.5 (`labs-leanstral-1-5`) with improved SFT mixture quality and extended long-context reasoning (retiring September 30, 2026); and Mistral Small Creative (`labs-mistral-small-creative`).

– Names model ids and retirement date but limited use guidance `snapshot-20260924`

09 **Codestral and Devstral coding model releases** NEW 60

Released Codestral 2508 (`codestral-2508`) and Codestral 2501 (`codestral-2501`), plus Codestral Embed (`codestral-embed`); and the Devstral coding-agent line: Devstral Small (`devstral-small-2505`), Devstral Small 1.1 (`devstral-small-2507`), Devstral Medium (`devstral-medium-2507`), Devstral 2 (`devstral-2512`), and Devstral Small 2 (`labs-devstral-small-2512`).

– Lists model ids without describing capability differences `snapshot-20260924`

THINNER COVERAGE BELOW

10 **Magistral reasoning model releases** NEW 53

Released Magistral Medium (`magistral-medium-2506`) and Magistral Small (`magistral-small-2506`), followed by Magistral Medium 1.1 (`magistral-medium-2507`) and Magistral Small 1.1 (`magistral-small-2507`), and Magistral Medium 1.2 (`magistral-medium-2509`) and Magistral Small 1.2 (`magistral-small-2509`).

– Names each version id but no capability details given `snapshot-20260924`

11 **Context biasing and diarization in Audio Transcriptions API** NEW 50

The Audio Transcriptions API added context biasing and a `diarize` parameter.

– Names the parameter but no mechanism or example `snapshot-20260924`

12 **Inline batch job creation** NEW 45

Released inline batching, enabling creation of batch jobs without file uploading.

– Describes behavior change but no endpoint/parameter name `snapshot-20260924`

13 **Specialized language and moderation models** NEW 30

Released Mistral Saba (`mistral-saba-2502`) and Mistral Moderation 2603 (`mistral-moderation-2603`).

– Names ids only, no description of capability `snapshot-20260924`

14 **Document Library API** NEW 28

Released a Document Library API to manage libraries.

– No endpoint names or parameters given `snapshot-20260924`

15 **Custom structured outputs for all models** NEW 25

Released custom structured outputs for all models.

– No schema, endpoint or parameter details given `snapshot-20260924`

16 **SDK support for Audio and Transcription** NEW 21

Added SDK support for Audio and Transcription.

– Bare mention with no specifics on SDKs or methods `snapshot-20260924`

17 **Agents API** NEW 21

Released the new Agents API (May 2025).

– No endpoints or capabilities described `snapshot-20260924`

└──▷ **BREAKING ON UPGRADE**

! The new `p` parameter added to Completion API streamed chunks may break applications relying on strict parsing of those chunks; users of the `mistral-common` package must update to version 1.8.4 or higher.

`snapshot-20260924` NOTES ↗ **LATEST** STABLE

Mistral snapshot-20260924 adds OCR 4.1 GA, block-level confidence scores, and `include_blocks` paragraph layout extraction.

Promotes OCR 4.1 (`mistral-ocr-4-1`) to Generally Available; `mistral-ocr-latest` and `mistral-ocr-4` now point to it.

Adds `"block"` granularity to the `confidence_scores_granularity` parameter in the OCR API: `"page"` returns page-level scores only, `"block"` returns page-level and block-level scores, and `"word"` returns page-level and word-level scores.

Introduces `include_blocks` in the OCR API: when set to `true` , each page returns a `blocks` array with paragraph-level bounding boxes and a structural label in reading order.

Releases Leanstral 1.5 (`labs-leanstral-1-5`), an updated Lean 4 formal proof engineering model with improved SFT mixture quality and extended long-context reasoning (retiring September 30, 2026).

Releases OCR 4 (`mistral-ocr-4-0`); `mistral-ocr-4` now points to it.

+39 more

└──▷ **BREAKING ON UPGRADE**

! The new `p` parameter added to Completion API streamed chunks may break applications relying on strict parsing of those chunks; users of the `mistral-common` package must update to version 1.8.4 or higher.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · September 24, 2026

Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue