THE AI TOOLCHAIN

NO. 061

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED OCTOBER 13, 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 104 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 21 tools

▾ // THE BRIEF HIDE

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

OPERATE

Stop losing an agent run when the process that owns it dies

vim ? help

Mastra 1.76.0 adds claim/recover semantics for agent run ownership plus atomic working-memory merges, so a half-finished run is reclaimed by another worker rather than left hanging or double-executed. Hayhooks 2.1.0 moves live tracing and durable execution history into a terminal UI with bounded execution indexes for Memory and Redis, so you debug a stuck pipeline from the same shell you deployed it in instead of hunting for a dashboard.

Mastra · hayhooks

BUILD

Put a human back in the loop on a long-running browser agent

Cloudflare's persistent browser sessions add Live View takeover and session recording, so a login wall or captcha becomes a hand-off rather than a failed run, and the recording is there when it fails anyway. Codex CLI's alpha.8 turns instant and code-mode interrupts on by default and adds policy-based file-contents integrity checks for sandboxed commands, so you can stop a run mid-edit and know what it touched.

Cloudflare Agents · OpenAI Codex CLI

DEPLOY

Pay for a long video once instead of on every token of it

Gemini's opt-in dynamic video processing cuts video input token usage by up to 88% for long-form content, which is the difference between a one-hour recording being a budget item and being a demo. AnythingLLM swaps Ollama for AMD's Lemonade runtime, putting GPU and NPU acceleration behind the built-in engine on AMD hardware that previously ran CPU-bound.

Gemini API · AnythingLLM

BUILD

◆ Model Training & Fine-Tuning

Unsloth AI Unsloth

Feedback ⓘ v0.1.907-beta NOTES ↗

RANK

Unsloth runs and fine-tunes language and diffusion models locally through a Python library and interface.

> Unsloth's v0.1.907-beta adds VS Code Copilot integration for locally served models, a batch of Unsloth Studio configuration/serving flags, new `/rewind` and `/compact` chat commands, per-quant GGUF model listing, and LTX-2.5 video generation with synchronized audio.

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

01 **Unsloth Studio configuration and serving options** NEW

85

Unsloth Studio gains several configuration additions in this release: the `--models-dir` flag to specify a local folder for model discovery, `--ssl-certfile` / `--ssl-keyfile` flags to serve Unsloth Studio over HTTPS from your own certificate, the `UNSLOTH_STUDIO_HOME` environment variable to keep `uv`, its cache, and the uv-managed Python inside a custom directory, and `UNSLOTH_STUDIO_KEEP_LEGACY=0` environment variable support through `docker/run.sh`.

— Names every flag and env var verbatim with exact usage. v0.1.907-beta

02 **Rewind and compact chat commands** NEW

75

`/rewind` (also available from a message's More menu) restores a conversation, tracked files in the chat folder, or both, with preview and undo. `/compact`, for saved GGUF or MLX chats with tool support, compacts context on the next message and marks compaction points with inline dividers.

— Describes behavior and scope of both commands in detail. v0.1.907-beta

03 **VS Code Copilot model picker integration** NEW

70

The new `unsloth start vscode` subcommand puts a locally served model into VS Code Copilot Chat's model picker, letting Copilot Chat use locally running Unsloth models directly.

— Exact command named but no further mechanism given. v0.1.907-beta

04 **Per-quant GGUF listing via models endpoint** IMPROVED

70

Each on-disk quantization of a multi-quant GGUF is now listed separately via `GET /v1/models`, making individual quant variants discoverable through the API.

— Names exact endpoint and method; limited further detail. v0.1.907-beta

05 **LTX-2.5 video generation with synchronized audio** NEW

65

Adds LTX-2.5 video generation from text or a start image with synchronized audio, including fp8/int8 loads from pre-quantized DiTs.

— Names model and quantization options but no usage steps. v0.1.907-beta

06 **Persistent logits return for FastModel inference** IMPROVED

60

The `UNSLOTH_RETURN_LOGITS=1` environment variable is now preserved across for_inference() calls on `FastModel`, so logits output stays enabled through repeated inference calls.

— Names env var and method but lacks broader context. v0.1.907-beta

WAS THIS USEFUL? 👍 👎

ModelScope MS-SWIFT

Feedback ⓘ v4.6.0 NOTES ↗ RANK

MS-SWIFT provides training, fine-tuning, reinforcement learning, evaluation, and deployment workflows for language and multimodal models.

> ms-swift v4.6.0 introduces a new `decision` task type for JEV-style models that return calibrated probabilities instead of text, adds support for six new model families including MiniCPM5 and DeepSeek-V4.1-Flash, and extends compatibility with newer transformers, vLLM, and TRL versions.

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

01 Decision task type for JEV-style models NEW

93

Adds the `decision` task type via `swift sft --task_type decision`, enabling JEV-style models to return calibrated probabilities over fixed option sets (`choice` / `score` / `noul`) in a single forward pass instead of generating text. Ships with ready-to-use recipes for three decision models — `JEV-27B-VL`, `Clef`, and `OmniJev` — covering training, inference, and LoRA merge/export, and any CausalLM can be converted using `--task_type decision`.

Convert any CausalLM into a JEV decision model that outputs calibrated probabilities over fixed option sets instead of free text — useful for evaluation or classification tasks.

```
$ swift sft --model <model_id_or_path> --task_type decision
```

— Names exact flag, option types, and three ready recipes v4.6.0

02 Support for six new model families NEW

75

Adds `MiniCPM5` (1B / 2B) LLM series with base and SFT variants and vLLM inference; `MiniCPM-V-4.7` (1B / 35B-A3B) vision-language models with image and video support; `GLM-5.3-Flash` with vision and video support; `DeepSeek-V4.1-Flash` with full-parameter SFT and GRPO on Megatron-SWIFT plus vLLM inference; `Xing4.0-29B-A4B` (MLA + MoE + mHC) with LoRA and full-parameter SFT (DDP / DeepSpeed ZeRO-3) and `grouped_mm` expert acceleration (~3.7x); and `MonkeyOCRv2` (S / B) OCR and document-understanding models.

— Names all six models and their training modes, lacks exact commands v4.6.0

THINNER COVERAGE BELOW

03 Compatibility with newer transformers, vLLM and TRL IMPROVED

51

Extends compatibility to `transformers` ≥ 5.17 kernel proxy, `vLLM` 0.28+ colocate weight sync / wake-up API, and TRL optional-dependency checks under Transformers 5.

— Names versions and APIs but no usage steps v4.6.0

04 **Moore Threads (MUSA) backend support** NEW 10

Adds a Moore Threads (MUSA) backend, as noted in the release summary, with no further detail provided.

— Mentioned only in summary with no elaboration v4.6.0

WAS THIS USEFUL?  

◆ AI Coding Agents

herdr

Feedback ⓘ v0.9.4 NOTES ↗ RANK

Herdr is a coding-agent runtime that manages persistent terminal sessions across local and SSH machines.

> Herdr's v0.9.4 cycle centers on plugin lifecycle management (`plugin update` , `plugin uninstall`) and a new `allow_unelevated_clients` config key/flag for sharing elevated Windows servers with ordinary clients, alongside several breaking changes to signal handling, Windows input, and default server access.

└──▷ **WHAT SHIPPED · 8 FEATURES** most completely described first ? what's the number?

01 Unelevated client access to elevated Windows servers

BREAKING

95

Adds an `allow_unelevated_clients` key in the `[server]` config section (shown as `server.allow_unelevated_clients: true` in YAML and `[server]` `allow_unelevated_clients = true` in TOML at `~/.config/herdr/config.toml`), plus a `--allow-unelevated-clients` flag on `herdr server` (also usable per-session as `herdr --session <name> server --allow-unelevated-clients`), letting an elevated Windows server accept connections from ordinary non-administrator clients on the same account. As of this change an elevated Windows server now only accepts administrator clients by default, so existing setups relying on normal-client access will be refused until this config key or flag is set.

Permanently allow ordinary-privilege clients to attach to an elevated herdr server on Windows, so a non-admin terminal in the same account can access administrator panes.

```ini
[server]
allow_unelevated_clients
```

Temporarily enable unelevated-client sharing for a single named elevated session without touching config — useful for one-off access grants that should not survive a restart.

```
$ herdr --session <name> server --allow-unelevated-clients
```

Persist cross-elevation access across restarts by setting the config key — useful when running herdr server as an elevated process and needing non-admin clients on the same account to reach administrator panes.

```yaml
server.allow_unelevated_clients: true
```

Enable cross-elevation access for a single elevated server session without changing the config file.

```
$ herdr server --allow-unelevated-clients
```

Allow non-administrator clients from the same account to connect to an elevated Windows herdr server.

```toml
[server]
allow_unelevated_clients = true
```

— Exact config key, flag, file path and formats all given. v0.9.4

02 **Plugin management CLI (new)** NEW 90

Adds `herdr plugin update` to refresh all GitHub-managed plugins or specific targets named by plugin id or `owner/repo[/subdir...]` source, preserving each install's requested ref, skipping unchanged commits, ignoring locally linked plugins during bulk updates, and supporting a noninteractive mode flag; plugin installs now build into a new Herdr-managed installation directory, leaving the previous installation in place during updates and reinstalls so running commands and panes keep their original files without a config reload or interruption; adds `herdr plugin uninstall`, accepting either the plugin ID or GitHub source, which preserves installation files until any running plugin invocations finish and the server exits; and adds support for `[<plugin_id owner/repo[/subdir...]>...]` arguments for noninteractive plugin installs.

Bulk-refresh all GitHub-managed plugins to their latest commits in one shot, skipping anything already up to date.

```
$ herdr plugin update
```

Update only specific plugins by id or GitHub source when you want targeted control over which plugins are refreshed.

```
$ herdr plugin update my-plugin-id owner/repo/subdir
```

Update all GitHub-installed plugins at once without restarting running panes.

```
$ herdr plugin update
```

Update a single named plugin by name while leaving locally linked plugins untouched.

```
$ herdr plugin update <name>
```

Remove a specific plugin by ID while letting any currently running invocations finish cleanly.

```
$ herdr plugin uninstall <plugin-id>
```

— Named command, arguments and update mechanics fully specified. Named command with runnable example and shutdown-safe mechanism. Argument syntax named but no usage context given.

v0.9.4

---

03 **SIGHUP no longer stops the server** BREAKING

75

On Linux and macOS, `SIGHUP` no longer stops the server; scripts that used `SIGHUP` to shut down Herdr must switch to `herdr server stop` or `SIGTERM`.

— Exact signal names and migration command given. v0.9.4

---

04 **Removal of HERDR_WINDOWS_INPUT_BACKEND override** DEPRECATED

60

The `HERDR_WINDOWS_INPUT_BACKEND=crossterm` environment variable is removed; Windows always uses its native console input.

— Names the removed variable and resulting behavior. v0.9.4

THINNER COVERAGE BELOW

---

05 **Fish shell reflow handling** IMPROVED

50

Sets `fish_handle_reflow=0` for Fish shell sessions since herdr handles text reflow on resize itself; an inherited or explicit `fish_handle_reflow` value set by the user is kept.

— Names the exact variable and override behavior. product docs

---

06 **WezTerm raw-paste text recovery on Windows** IMPROVED

45

Windows clients now recover text when WezTerm wraps a complete sequence of Win32 character press/release pairs in raw paste markers.

— Specific scenario named but no reproduction steps. product docs

07 **Confirmation prompt before closing last tab or pane** NEW

45

Adds a TUI confirmation prompt before closing a workspace's last tab or its only pane via a keybinding or the tab/pane menu; ordinary pane closes and CLI/API closes remain immediate.

— Describes scope and exceptions but no named flag. product docs

08 **Persistent clickable Windows notifications** NEW

30

Windows notifications now persist in Notification Center; clicking one jumps to its pane and brings the terminal window forward.

— Behavior described but no surface or setting named. v0.9.4

└──▷ **BREAKING ON UPGRADE**

! An elevated Windows server now only accepts administrator clients by default; existing setups that relied on normal clients connecting will be refused until `[server] allow_unelevated_clients = true` is set or the server is started with `--allow-unelevated-clients`.

! On Linux and macOS, `SIGHUP` no longer stops the server; scripts that used `SIGHUP` to shut down Herdr must switch to `herdr server stop` or `SIGTERM`.

! The `HERDR_WINDOWS_INPUT_BACKEND=crossterm` environment variable is removed; Windows always uses its native console input.

WAS THIS USEFUL? 👍 👎

Cline

Feedback ⓘ desktop-v0.0.46 NOTES ↗ CODE ↗ RANK

Cline is an open-source AI coding agent for VS Code that can read and edit files, run terminal commands, and use browsers with user approval.

> Cline's latest release reworks default model selection across many providers and gives the `ai&` provider its own dedicated setup flow, alongside breaking changes to the `v0` and OpenCode Go providers.

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

01 **Default model changes across providers** UPDATED INTEGRATION NEW INTEGRATION BREAKING 75

Claude Sonnet 5.5 is now the default model for the Cline and OpenRouter providers when no model is explicitly chosen. This also changes default behavior for many other providers that previously had different defaults: users of 302.AI, Cortecs, CrossModel, DevPass, DigitalOcean, Eden AI, GitHub Copilot, Google Vertex AI, LLM Gateway, Meta, Moark, NanoGPT, NEAR AI Cloud, Neuralwatt, Novita AI, Ofox, OpenCode Go, Requesty, Vivgrid, and several others who have not pinned a model will see a different default after upgrading, with most switching to Claude Haiku 5.5 or Muse Wonder 1.0.

– Names exact new defaults and affected providers, but no pinning steps shown v4.1.24

02 **v0 provider limited to single default model** BREAKING 60

The `v0` provider now offers only its default model `v0-1.5-md`; any model other than `v0-1.5-md` that was previously pinned will no longer be available.

– Names exact model id affected but no migration instructions v4.1.24

03 **OpenCode Go endpoint change** UPDATED INTEGRATION BREAKING 60

OpenCode Go now routes requests to `https://opencode.ai/inference/go/v1`; any configuration pointing to the old endpoint will stop working.

– Gives exact new endpoint but no steps to update existing config v4.1.24

THINNER COVERAGE BELOW

04 **Dedicated setup for ai& provider** IMPROVED 45

Selecting the `ai&` provider now shows its own dedicated API key and model picker instead of the generic OpenAI-compatible form with base URL and custom headers.

– Describes the UI change but gives no exact navigation path v4.1.24

NEW INTEGRATION AI/ML API provider

└──▷ BREAKING ON UPGRADE

! If you use 302.AI, Cortecs, CrossModel, DevPass, DigitalOcean, Eden AI, GitHub Copilot, Google Vertex AI, LLM Gateway, Meta, Moark, NanoGPT, NEAR AI Cloud, Neuralwatt, Novita AI, Ofox, OpenCode Go, Requesty, Vivgrid, or several other providers without pinning a model, expect a different default model after upgrading (most change to Claude Haiku 5.5 or Muse Wonder 1.0).

! The `v0` provider now offers only its default model `v0-1.5-md`; any pinned model other than `v0-1.5-md` will no longer be available.

! OpenCode Go now routes to `https://opencode.ai/inference/go/v1`; any configuration pointing to the old endpoint will stop working.

desktop-v0.0.46 NOTES ↗ CODE ↗ LATEST STABLE

Adds a conversation outline, plugin error status lines, `.claude/skills` discovery, improved scheduled routine reliability, and cross-provider reasoning toggle.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch desktop-v0.0.46
https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout desktop-v0.0.46
```

Adds a conversation outline along the left edge of the chat: hover a mark to preview the prompt, click to jump to it.

Plugins that fail to load or error during a session now show a named status line in chat identifying the plugin and the error.

Discovers skills placed in `.claude/skills` automatically, in addition to the existing skills directory.

Scheduled routines now mark failed model calls (e.g. a retired model) with an error, run only the latest missed occurrence after downtime, and the routine editor lists current models instead of retired ones.

Turning reasoning off now works across every provider; level-only Gemini models no longer reject the request with a thinking-budget error.

+3 more

cli-v3.0.71 NOTES ↗ CODE ↗ STABLE

Cline CLI v3.0.71 loads plugins in-hub (no JS runtime needed on PATH), adds `cline hub plugins` with `--reload`, `CLINE_PLUGIN_MODE=sandbox`, and enables `--thinking` in ACP mode.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch cli-v3.0.71 https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout cli-v3.0.71
```

Adds `cline hub plugins` subcommand to show each plugin's status and last error, with `--reload <plugin>` to re-import a single plugin without restarting the hub

Adds `CLINE_PLUGIN_MODE=sandbox` environment variable to preserve the old subprocess-based plugin behavior

Adds `plugin_failure` entry type to the output of `cline config tools --json` when a plugin fails to load

`--thinking` now applies in ACP mode (`--acp`), enabling reasoning in Agent Client Protocol sessions

`cline hook` now accepts `agent_error` events instead of exiting with an error and no response

+3 more

WAS THIS USEFUL? 👍 👎

Token Optimizer

Feedback ⓘ v5.13.40 NOTES ↗ RANK

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

> Token Optimizer's single release in this window introduces a Devin CLI beta with lossless `git log` compression, credential-safe colour output, pre-run integrity checks, and more resilient install/upgrade lifecycle handling.

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

01 **Resilient install and upgrade lifecycle for Devin** IMPROVED 78

Adds resilient lifecycle management for Devin: `install`, `refresh`, `uninstall`, and `purge` operations survive a moved or deleted skill folder, an interrupted install, or an upgrade from 5.13.39 without removing user-edited or user-added files.

– Names all four subcommands and failure scenarios covered v5.13.40

02 **Pre-run integrity check with doctor** NEW 66

Adds a pre-run integrity check on Devin: the filter validates every installed file against its install record before executing, and `doctor` confirms the full payload loads.

– Names `doctor` command as actionable entry point v5.13.40

03 **Lossless git log compression for Devin CLI** NEW 60

Adds a Devin CLI beta output-filter that compresses `git log` output roughly 10x with byte-for-byte recovery.

– Names mechanism and ratio but no invocation flag v5.13.40

THINNER COVERAGE BELOW

04 **Credential-safe colour output on Devin** NEW 50

Preserves credentials in raw form on Devin even when output is colour-highlighted, such as from `grep --color=always`.

– Clear behaviour but no command to invoke v5.13.40

WAS THIS USEFUL? 👍 👎

yetone magpie

Feedback ⓘ v0.1.1228 CODE ↗ RANK

Magpie is a macOS menu-bar app, TUI, and CLI that lists every AI coding agent on the machine and switches each one's model, such as Codex on DeepSeek or Claude Code on Kimi, by rewriting only that key in the agent's own config.

> Magpie shipped a new plugin kernel that lets plugins draw their own pages (starting with the Gateway page) and gained a built-in plugin toggle, added daemon service management so the gateway can run as a persistent systemd/launchd service, and expanded gateway reliability (quota handling, retry logic, key scoping) alongside new sync targets, account types, and agent integrations.

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

01 **Daemon service management** NEW 88

`magpie daemon install|uninstall|start|stop|restart|status` subcommands manage the gateway as a systemd user unit on Linux or a launch agent on macOS, letting it survive reboots and SSH disconnects.

Install and start magpie as a persistent background service so the gateway survives SSH disconnects and reboots.

```
$ magpie daemon install
magpie daemon start
magpie daemon status
```

— Full subcommand set with mechanism and a runnable example v0.1.1227

02 **Plugin system: kernel, pages, and built-in toggle** NEW 85

magpie's UI gains a plugin kernel (`kernel.js`) that lets plugins register pages via slots.add('page', part), wrapped in an error boundary so a broken plugin page doesn't crash other pages; the Gateway page is now the first page drawn through this kernel, using its `mount` , `loading` , `draft` , `render` hooks instead of being built into the shell. Plugins can render settings forms via `ctx.ui` and `ui.form` , driven by the Go-side `pluginrt.SchemaOf` schema. A new Plugins › Built-in UI section lets you enable or disable built-in plugins individually, powered by the `pluginrt` runtime which reads `plugins.yml` and resolves plugin dependency order via Requires and Provides .

— Names kernel API, hooks and runtime config but only a UI path, no command v0.1.1228 v0.1.1225

03 **GitHub repository sync for encrypted backups** NEW 81

Adds GitHub repository as a sync target for encrypted settings backups alongside WebDAV and S3, addressed as `github://owner/repo` or `github://owner/repo/folder`; the repository must be private.

Sync your encrypted magpie settings to a private GitHub repository instead of WebDAV or S3.

```text
github://owner/repo
```

— Exact address format and constraint, with example v0.1.1225

04 **Instruction-file library** NEW 78

Agents can share a common prompt library stored outside any single project via instruction sets kept in a file such as `~/.agents/AGENTS.md`. Library → Instructions gained entries for Kimi Code, fx, Prime Agent, StepCode, jcode, Qoder, Command Code, CodeBuddy Code, and Reasonix, each pointing to the file its source reads.

Keep a shared system prompt for all agents in a file outside any single project so every agent run picks it up automatically.

```
$ mkdir -p ~/.agents
echo '# Global agent instructions
Always confirm destructive actions.' > ~/.agents/AGENTS.md
```

— Names the file path and all nine library entries, with a runnable example v0.1.1227 v0.1.1225

05 **Codex web search endpoint** NEW 73

Adds `AlphaSearchPath` (`/alpha/search`) endpoint so Codex's `web.run` tool can execute web searches against magpie's models without receiving a fatal 404.

— Names endpoint and fixed failure mode, no example call shown v0.1.1226

06 **Bare model ID listing** IMPROVED 71

Exposes each model by its bare `id` in `GET /v1/models` to support apps migrated from another proxy.

List models by bare ID so a client originally pointed at another proxy can resolve model names without aliasing.

```
$ curl -s 'http://localhost:9090/v1/models' | jq '.data[].id'
```

— Named endpoint with a runnable curl example v0.1.1226

07 **Gateway quota handling, retry logic and key scoping** IMPROVED

70

Gateway rests a model on quota when it returns a 402 usage-limit response until the billing window renews; retries a tool call without the `strict` parameter when an upstream refuses it, omitting `strict` on subsequent calls to that upstream; and scopes gateway key budget/model/account info so a key bound to a budget, models, or accounts sees only its own limit and routes, not all accounts.

— Names three distinct mechanisms, no CLI step to invoke v0.1.1227

08 **Codex auto-use reset control** NEW

68

A Settings › Usage toggle turns each Codex account's auto-use resets on and off independently, and the last reset magpie used (when and why) is now shown on a Codex account's card in the provider, GUI, and CLI views.

— Exact settings path and surfaced fields, no command given v0.1.1227

THINNER COVERAGE BELOW

09 **Sandbox desktop-state env var clearing** IMPROVED

55

Every sandbox now clears `PI_DESKTOP_DATA_DIR` so service tools run through `proc` without leaking desktop data paths, and clears `REASONIX_STATE_HOME` so Library reads Reasonix's `REASONIX.md` from the correct location.

— Names both env vars and their purpose, no user action needed v0.1.1227 v0.1.1225

10 **Configurable reasoning effort for vision/search models** NEW

52

Enables configurable reasoning effort via `VisionEffort` and `SearcherEffort` settings for Image recognition and Web search models.

— Names both settings but no values or location given v0.1.1226

11 **New agent integrations for skills and MCP servers** NEW

52

A project's skills and MCP servers can now reach Qoder, CodeBuddy, Droid, Cline, and Command Code.

— Names five new agents, no setup steps given v0.1.1225

12 **Check for Updates using GitHub token** IMPROVED 52

Check for Updates now uses the GitHub token from Settings to authenticate GitHub API requests, and surfaces GitHub rate-limit exhaustion on a per-plugin basis.

— Names mechanism and settings source, no exact steps v0.1.1225

13 **Partial repository checkout for skills** NEW 52

Enables partial repository checkout for skills so a skill can install from a repository too large to download in full, by fetching only the relevant skills' folders.

— Explains mechanism, no config key or command named v0.1.1225

14 **Sub2API Claude account support** NEW 45

Adds Sub2API Claude accounts, identified by `email_address` with `account` and `organization` fields.

— Names the fields but no setup steps shown v0.1.1227

15 **MLX-Serve decision model routing** IMPROVED 44

MLX-Serve decision models are now classified and routed through System One as their own routing group, separate from chat models.

— Explains new routing split, no config surface named v0.1.1226

16 **Menu-bar usage display window** IMPROVED 42

Menu-bar token/cost display is now selectable between today's usage and the last 7 days.

— States the toggle but no exact UI path v0.1.1227

17 **Privacy disclosure page** NEW 30

A Privacy page lists everything magpie sends and how to disable it.

— Bare description, no specifics on what's sent v0.1.1227

WAS THIS USEFUL? 👍 👎

OpenAI Codex CLI

Feedback ⓘ rust-v0.163.0-alpha.8 NOTES ↗

CODE ↗ RANK

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

> Codex CLI's alpha.8 release adds policy-based file-contents integrity checks for sandboxed commands, turns on instant and code-mode interrupts by default, and adds a Windows-specific sandbox opt-out alongside experimental prediction forks and model-specific function prefixes.

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

01 **Policy-based file-contents integrity checks** NEW 55

Adds policy-based file-contents integrity checks with bubblewrap (Linux) and macOS Seatbelt backends, which run before commands and filesystem operations to verify file contents against policy.

— Names backends and timing but no config/flag to control it rust-v0.163.0-alpha.8

02 **Windows MXC sandbox opt-out** NEW 55

Adds a Windows MXC sandbox opt-out that preserves an MXC backend selected via CLI override across child processes.

— Names platform, backend, and CLI override mechanism rust-v0.163.0-alpha.8

03 **Experimental prediction forks** NEW 30

Adds experimental prediction forks that inherit parent context.

— Names the feature but no mechanism or usage detail rust-v0.163.0-alpha.8

04 **Instant interrupts on by default** IMPROVED 25

Enables instant interrupts and code-mode interruption by default, so agent execution can be stopped more immediately during both normal and code-mode operation.

— Behavior change stated but no flag or command shown rust-v0.163.0-alpha.8

05 **Model-specific function description prefixes** NEW 25

Adds support for model-specific function description prefixes.

— Bare feature name with no mechanism or example rust-v0.163.0-alpha.8

06 **Thread/list exclusions** NEW 10

Release summary mentions thread/list exclusions as part of this version, though no further detail on scope or configuration is provided.

— Only named in summary, no elaboration given `rust-v0.163.0-alpha.8`

WAS THIS USEFUL? 👍 👎

Graphify Labs Graphify

Feedback ⓘ v0.9.89 NOTES ↗

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

Adds interface-dispatch fan-out to all implementers, improves agent hook nudges to surface graph nodes, and bumps AST cache schema (re-extract required on upgrade).

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.9.89 https://github.com/Graphify-Labs/graphify.git
# already have the repo? check out this version:
$ git checkout v0.9.89
```

Adds interface-dispatch fan-out so an interface member now links to every implementer (not just one), enabling DI chains through multi-implementer interfaces to resolve correctly.

Improves read/search hook nudges to surface the graph nodes behind the nudged call and tighten wording so agents act on them rather than treating them as noise.

NEW SUPPORTED PLATFORM Cross-repo member-call resolution for Go, PHP, TypeScript/JS, Kotlin, Objective-C

NEW SUPPORTED PLATFORM Objective-C NS_ENUM and NS_OPTIONS enum extraction

NEW SUPPORTED PLATFORM Pi shell hook-guard integration

└──▷ BREAKING ON UPGRADE

! The AST cache schema is bumped — the first build after upgrading will re-extract all files from scratch.

WAS THIS USEFUL? 👍 👎

Amazon Kiro

Feedback ⓘ

General: Kiro is now in scope for FedRAMP Class C (2026-10-12) BLOG ↗

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

NEW COMPLIANCE FRAMEWORK FedRAMP Class C

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

OpenClaw

Feedback ⓘ v2026.10.3 NOTES ↗ CODE ↗ RANK

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

> OpenClaw v2026.10.3 expands model support with Muse Wonder and adds thinking-effort controls for MiniMax M3.1.

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

01 **MiniMax M3.1 thinking-effort controls** IMPROVED 40

MiniMax M3.1 models now honor the selected thinking effort level.

— Names model and setting but no mechanism or UI path v2026.10.3

02 **Muse Wonder model support** NEW 30

OpenClaw adds Muse Wonder as an available model.

— Names the model but no setup detail v2026.10.3

WAS THIS USEFUL? 👍 👎

Cloudflare Agents

Feedback ⓘ agents@0.28.1 NOTES ↗ RANK

The Cloudflare Agents SDK builds stateful agents on Workers and Durable Objects, with persistent state, scheduling, WebSockets, and MCP server support.

> Cloudflare Agents adds persistent, human-supervisable browser sessions—complete with Live View takeover, session recording, and multi-harness adapters—alongside zero-data-retention AI Gateway routing, dynamic gateway routes, and a unified OpenCode session event stream.

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

01 Persistent browser sessions across agent turns NEW

95

browserRun(binding, options) accepts a `keepAliveMs` option (idle default: 10 minutes) to keep a remote Cloudflare Browser Run session alive between agent turns, preserving tabs, cookies, and login state across Durable Object hibernation. A `browserTool` adapter importable from `agents/browser/pi`, `agents/browser/ai-sdk`, or `agents/browser/tanstack-ai` wires the persistent browser into Pi Durable, AI SDK (`generateText` / `streamText`), or TanStack AI (`chat`) harnesses; a `scope` parameter on `browserTool` (and Browser.connect({ scope }) for custom harnesses) gives each conversation its own active tab, with an omitted `scope` sharing one tab across all conversations in the AI SDK and TanStack AI adapters. A `timeoutMs` option on browser tool calls (default 60 seconds) bounds interrupted or long-running operations, and a `sessionId: 'active'` CDP command value lets a model return to its active tab within a conversation.

Give each user conversation its own isolated active tab when using the AI SDK adapter, so parallel sessions do not share browsing state.

```javascript
import { browserTool } from 'agents/browser/ai-sdk';

const tools = {
  ...browserTool(this.browser, { scope: conversationId }),
};
// Pass tools to generateText() or streamText().
```

— Names exact imports, options and defaults; runnable example given product docs

02 Live View human takeover for browser sessions NEW

85

browser.liveView() returns per-tab `liveViewUrl`, `pageUrl`, `title`, `sessionId`, and `expiresInMs` so a human can watch or take over the agent's browser in real time for logins, MFA, or CAPTCHAs. browser.close() explicitly ends a Browser Run session, after which any scope that subsequently uses the browser receives `restarted: true` in its tool result; browser.resolve() obtains the active Browser Run session ID while a browser is open, enabling post-close retrieval of recordings via getBrowserRecording() through the Browser Run REST API.

Hand off a stuck agent to a human for MFA or CAPTCHA by surfacing Live View links from the Durable Object, then letting the model continue in the same authenticated browser on the next turn.

```javascript
getLiveView() {
  const view = await this.browser.liveView({ mode: 'devtools' });
  if (!view) return [];
  return view.targets
    .filter((tab) => tab.pageUrl?.startsWith('http'))
    .map((tab) => ({
      title: tab.title,
      pageUrl: tab.pageUrl,
      liveViewUrl: tab.url,
    }));
}
```

— Exact method/field names and a working handoff example `product docs`

---

03 **Zero-data-retention and dynamic AI Gateway routing** NEW

80

A `zdr` gateway option and `vendorKey: "request"` were added to `createAI` in `agents/models/ai-sdk` and `agents/models/pi-ai` for zero-data-retention routing and bring-your-own-vendor-key flows. Dynamic AI Gateway route support was also added to `createAI` via the ai("dynamic/<route>") syntax in the same modules.

Route AI Gateway calls through a zero-data-retention endpoint with your own vendor key instead of Cloudflare billing.

```typescript
import { createAI } from 'agents/models/ai-sdk';

const ai = createAI({
  gateway: { id: 'my-gateway', zdr: true },
  vendorKey: 'request'
});
```

Send model traffic through a dynamic AI Gateway route that is resolved at request time rather than at configuration time.

```typescript
import { createAI } from 'agents/models/ai-sdk';

const ai = createAI({});
const model = ai('dynamic/my-dynamic-route');
```

— Exact option names with runnable configuration examples `agents@0.28.1`

04 **Unified OpenCode session event stream** IMPROVED 75

`OpenCodeHarness` 's session.events() now returns a unified stream with a snapshot plus translated events (`run_start` / `run_end` , `operation_start` / `operation_end` , `message_start` / `message_end` , `text_delta` , `reasoning_delta` , `tool_start` / `tool_end` , permissions, compaction) matching the kinds `PiHarness` reports, while raw durable events remain accessible from session.log(). The OpenCode example now provides the model with Workspace tools from `@cloudflare/computer` instead of OpenCode's local file and shell tools.

Consume a normalised, resumable event stream from an OpenCode session — compatible with the same event kinds PiHarness emits.

```typescript
const stream = session.events();
for await (const event of stream) {
  if (event.type === 'text_delta')
process.stdout.write(event.delta);
  if (event.type === 'tool_start') console.log('tool:',
event.name);
  if (event.type === 'run_end') break;
}
```

— Named event types and a runnable consumption example `agents@0.28.1`

05 **Browser session recording** NEW 60

A `recording: true` option on browserRun() provider options captures Browser Run sessions as structured rrweb events (DOM changes, input, navigation) per tab, retained for 30 days and capped at two hours per session.

— Clear flag and limits but no retrieval example here `product docs`

06 **Lifecycle class and SQLite-backed browser state** NEW 60

A `Lifecycle` class (imported from `agents/lifecycle`) manages shared startup and shutdown of `Browser` and `PiHarness` components within a Durable Object, and the SDK supports the `new_sqlite_classes` Durable Object migration tag in `wrangler.toml` / `wrangler.json` for the `BrowserAssistant` class, enabling SQLite-backed persistent browser state.

— Names class, import path and config tag but no usage example `product docs`

THINNER COVERAGE BELOW

07 **MCP session termination hardening** IMPROVED 40

`McpAgent` now immediately returns 404 for requests on sessions ended with `DELETE`, and `createMcpHandler` rejects a Legacy request whose factory returns an instance still serving another request.

— Describes behavior change but gives no example or migration steps `agents@0.28.1`

WAS THIS USEFUL? 👍 👎

Feynman

Feedback ⓘ v0.5.44 NOTES ↗ RANK

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

> Feynman expanded its web search backend to support all 39 pi-web-access providers and added a same-origin redirect-safety setting for content fetching.

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

01 Search provider support expanded to 39 pi-web-access providers

NEW 90

`feynman search set` now accepts all 39 providers supported by `pi-web-access` — including Brave, Tavily, Kagi, Serper, You.com, SearXNG, and DuckDuckGo — up from only `exa`, `perplexity`, and `gemini`. `feynman search status` and `feynman doctor` were updated to name the configured provider and list every saved key, replacing the outdated automatic search-order description.

Switch your active search provider to Kagi so all web lookups route through it instead of the previous default.

```
$ feynman search set kagi
```

Switch your web search backend to Kagi so all research workflows use it instead of the previous default providers.

```
$ feynman search set kagi
feynman search status
```

— Names exact command, providers, and companion status commands v0.5.44

02 Same-origin redirect policy for content fetching NEW

75

Adds `fetchContent.redirects: "same-origin"` as an opt-in setting, introduced in pi-web-access 0.40.0, to refuse content-fetch redirects to a different origin — useful for scoping research to trusted or non-hostile sources.

Prevent content fetches from silently following redirects to external domains — useful when scoping research to trusted origins.

```yaml
fetchContent.redirects: "same-origin"
```

Lock down web fetching so pi-web-access refuses cross-origin redirects — useful when researching potentially hostile sources.

```yaml
fetchContent:
  redirects: "same-origin"
```

— Exact config key and value given, with version it shipped in `v0.5.44`

WAS THIS USEFUL? 👍 👎

Mastra

Feedback ⓘ @mastra/core@1.76.0 NOTES ↗ RANK

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

> Mastra 1.76.0 ships durable agent run ownership with claim/recover semantics, atomic working-memory merges, trace nesting, new file-handling processors, and a model reasoning-options query, alongside a docs-only Metrics API exposed via CLI, SDK, and HTTP.

└──▷ **WHAT SHIPPED · 11 FEATURES** most completely described first ? what's the number?

01 **Atomic merge for working memory** NEW 95

Adds memory.mergeWorkingMemory() and client.updateWorkingMemory() with `mode: 'merge'` to deep-merge schema-based working memory atomically inside a row-locked PostgreSQL transaction using `scope: 'resource'`; adapters that don't support this return HTTP `400` instead of silently overwriting concurrent writes.

Merge a partial working-memory update into Postgres without overwriting other concurrent writers' fields.

typescript

```
await memory.mergeWorkingMemory({
  threadId: 'thread-123',
  resourceId: 'user-456',
  workingMemory: { city: 'Berlin' }
});
```

— Full mechanism, exact method names and runnable example @mastra/core@1.76.0

02 **Durable agent run ownership and recovery** NEW 93

Adds durable agent run ownership: stream(), generate(), resume(), and recover() now claim a run; recoverActiveRuns() skips live-claimed runs (returning `status: 'skipped'`, a `reason`, and `retryAt`); recover() throws `DURABLE_AGENT_RUN_ACTIVE` for live runs, accepting `force: true` to take over, and throws `DURABLE_AGENT_RECOVER_RUN_ACTIVE_LOCALLY` for same-process conflicts; mastra.recoverAllDurableAgents() adds a `skipped` count; mastra.shutdown() releases runs after the drain window; storage adapters can enforce run fences to reject writes from executions that lost ownership.

— Exhaustive named APIs and error codes, but no runnable example @mastra/core@1.76.0

03 **Nest agent runs under parent trace** NEW 86

Adds `tracingOptions.nestUnderParent` and the nestUnderRun() helper (from `@mastra/core/observability`) to attach a run as a true child span under an earlier run in the same trace, preserving the parent run's name, input, output, and tags.

Score an LLM response with a judge agent nested under the original run in the same trace for clean observability.

typescript

```
import { nestUnderRun } from '@mastra/core/observability';

const turn = await assistant.generate('When does my order
ship?');

await judge.generate('Grade this answer', {
  tracingOptions: nestUnderRun(turn)
});
```

— Exact import path and helper with runnable example `@mastra/core@1.76.0`

04 **File upload and error-handling processors** NEW 78

Adds `FileUploadProcessor`, an input processor that uploads user-sent files to the workspace sandbox and passes the model a sandbox path instead of raw file content, with a per-file `filter` function to exclude files like images. Adds `isMalformedFunctionCallError` (from `@mastra/core/processors`) for use in custom `StreamErrorRetryProcessor` matchers to handle Gemini's intermittent `MALFORMED_FUNCTION_CALL` finish reason. Adds `UnsupportedFileHandler`, a new default error processor that retries a call with the rejected file removed or replaced by a sandbox path when a provider rejects an attached file, so affected threads resume on the next turn.

— Names all three processors but no runnable example `@mastra/core@1.76.0`

05 **Query model reasoning capabilities** NEW 78

Adds getModelReasoningOptions() (from `@mastra/core/llm`) to query which reasoning controls a model accepts — named effort levels, a thinking-token budget, or an on/off toggle — as published by models.dev.

Check which reasoning controls a model supports before offering reasoning options in your UI.

```typescript
import { getModelReasoningOptions } from '@mastra/core/llm';

const options = getModelReasoningOptions('anthropic/claude-sonnet-4-6');
// [{ type: 'effort', values: ['low', 'medium', 'high', 'max'] },
{ type: 'budget_tokens', min: 1024 }]
```

— Exact function, import path and example provided `@mastra/core@1.76.0`

06 **Metadata on delegated sub-agent threads** NEW 70

Adds `threadMetadata` to the `onDelegationStart` return value so delegated sub-agent threads carry metadata (e.g. `tenantId`) that makes them findable via memory.listThreads() with `filter: { metadata: ... }`.

— Names exact fields and filter syntax, no example `@mastra/core@1.76.0`

07 **Metrics API via CLI, SDK, and HTTP** NEW 70

Adds a Metrics API for querying exported observability metrics via the `mastra api metric` CLI subcommand, `MastraClient` SDK methods, and a new HTTP API — all supporting regional endpoints, authentication, and project scoping.

Query exported observability metrics from the CLI without writing any code — useful in CI or ops scripts.

```
$ mastra api metric
```

— Exact CLI command given; SDK/HTTP details stay generic `product docs`

08 **Metrics UI components rebuilt in playground** BREAKING 68

Old Metrics card views/components in `@mastra/playground-ui` were removed in favor of new self-contained Metrics cards/dashboard building blocks; `MetricsCardGroup` no longer accepts the `variant` prop, and `MetricsLineChart` dropped the `xAxisInterval` and `xAxisMinTickGap` props.

— Names exact removed props and components needing migration `@mastra/core@1.76.0`

09 **Observational memory default model changed** BREAKING 68

`@mastra/memory` Observational Memory default changed from `model: 'google/gemini-2.5-flash'` to `model: 'auto'` for `model`,

`observation.model`, and `reflection.model` — a different provider or model may be selected depending on available credentials.

— Exact before/after config values given `@mastra/core@1.76.0`

THINNER COVERAGE BELOW

10 **Trace querying operators and metrics usageId field** IMPROVED 52

Trace querying gains `matches` and `notMatches` word/phrase operators across stores and types, and metrics add a `usageId` field to group token/cost rows per model call.

— Named operators and field but no usage example `@mastra/core@1.76.0`

11 **Workspace access in processor context** NEW 48

Exposes `processLLMRequest` processor context with `workspace` so processors can interact with the agent's sandbox or filesystem before a model call.

— Brief description, no mechanism or example `@mastra/core@1.76.0`

└──▷ **BREAKING ON UPGRADE**

! `@mastra/playground-ui`: Old Metrics card views/components were removed in favor of new self-contained Metrics cards/dashboard building blocks.

! `@mastra/playground-ui`: `MetricsCardGroup` no longer accepts the `variant` prop; `MetricsLineChart` dropped the `xAxisInterval` and `xAxisMinTickGap` props.

! `@mastra/memory` Observational Memory default changed from `model: 'google/gemini-2.5-flash'` to `model: 'auto'` for `model`, `observation.model`, and `reflection.model` — a different provider or model may be selected depending on available credentials.

WAS THIS USEFUL? 👍 👎

◆ MCP Servers & Tooling

hayhooks

Feedback ⓘ v2.1.0 NOTES ↗ RANK

Easily deploy Haystack pipelines as REST APIs and MCP Tools.

> Hayhooks 2.1.0 replaces its browser-based dashboard with a terminal UI for live tracing and durable execution history, locks down tracing/monitoring behind authentication and defaults, and adds bounded execution indexes for Memory and Redis stores.

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

01 Terminal dashboard (TUI) command NEW

95

Adds the `hayhooks dashboard` TUI command for browsing server health, recent executions, and paginated durable execution history from the terminal, with keyboard shortcuts `d` (deployments), `e` (executions), and `v` (traces). Also supports `hayhooks dashboard --once --json` for a noninteractive, scriptable status snapshot, pairable with `--pipeline NAME` for durable executions or `--traces` for local traces; requires no Node.js or browser and installs via `pip install 'hayhooks[tui]'`.

Inspect live pipeline traces and execution history from the terminal without a browser or Node.js.

```
$ pip install --upgrade 'hayhooks[tui]==2.1.0'
hayhooks run &
hayhooks dashboard
```

Emit a one-shot JSON status snapshot of durable executions for a specific pipeline — useful in CI or monitoring scripts.

```
$ hayhooks dashboard --once --json --pipeline my_agent
```

— Full command, flags and install step given v2.1.0

02 Tracing and monitoring defaults and config renamed BREAKING

85

Local tracing capture and durable monitoring indexes are now enabled by default; disable with the `--without-tracing` / `--without-monitoring` server flags or `HAYHOOKS_TRACING_ENABLED=false` / `HAYHOOKS_MONITORING_ENABLED=false` env vars. Server capture env vars are renamed from `HAYHOOKS_DASHBOARD_TRACE_*`

to `HAYHOOKS_TRACING_*`, and the slow-component setting is now `HAYHOOKS_TRACING_SLOW_COMPONENT_MS`; obsolete frontend and trace-stream settings are removed.

— Exact flags and renamed env vars given v2.1.0

---

03 **Authenticated GET /traces endpoint replaces old tracing API**

80

BREAKING

The local tracing API is now an authenticated `GET /traces` endpoint returning bounded, paginated snapshots. The previous tracing config, entrypoint, SSE stream, server-clear APIs, and incremental cursor parameters are removed, so API clients must poll bounded snapshots instead.

— Endpoint named but no example call shown v2.1.0

---

04 **React browser dashboard removed** BREAKING

75

The React browser dashboard, the `/dashboard` route, `--with-tracing-dashboard` and its aliases, and `--dashboard-path` are removed; use `hayhooks dashboard --traces` for terminal-based trace viewing instead.

— Clear migration path to the replacement command v2.1.0

---

05 **Operator authentication tokens for dashboard and tracing** NEW

70

Adds `HAYHOOKS_TRACING_TOKEN` (server) and `HAYHOOKS_DASHBOARD_TOKEN` (client) env vars so operators must explicitly authenticate when connecting the dashboard to remote or Windows servers.

Connect a dashboard client to a remote Hayhooks server that requires explicit operator authentication.

```
$ HAYHOOKS_TRACING_TOKEN=secret hayhooks run
# On the client machine:
HAYHOOKS_DASHBOARD_TOKEN=secret hayhooks dashboard --traces
```

— Named env vars with a runnable example v2.1.0

06 **Durable execution indexes for Memory and Redis** NEW

60

Adds bounded, owner-scoped execution indexes for Memory and Redis stores, with pagination and retention-aware cleanup, backing the durable execution history view in the dashboard.

— Describes mechanism but no config surface named v2.1.0

└──▷ **BREAKING ON UPGRADE**

! The React browser dashboard, the `/dashboard` route, `--with-tracing-dashboard` and its aliases, and `--dashboard-path` are removed; use `hayhooks dashboard --traces` instead.

! The local tracing API is now authenticated `GET /traces`; the old tracing config, entrypoint, SSE stream, server-clear APIs, and incremental cursor parameters are removed — API clients must poll bounded snapshots.

! Local tracing capture and durable monitoring indexes are now enabled by default; disable with `--without-tracing` / `--without-monitoring` or `HAYHOOKS_TRACING_ENABLED=false` / `HAYHOOKS_MONITORING_ENABLED=false`.

! Server capture env vars are renamed from `HAYHOOKS_DASHBOARD_TRACE_*` to `HAYHOOKS_TRACING_*`; the slow-component setting is now `HAYHOOKS_TRACING_SLOW_COMPONENT_MS`; obsolete frontend and trace-stream settings are removed.

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Agentic Infrastructure

Hugging Face

Feedback ⓘ changelog · Oct 13, 2026 CHANGELOG ↗

RANK

Hugging Face is an open platform for building, sharing, and deploying machine-learning models, datasets, and applications.

> Hugging Face shipped a scoped contributor access-token permission for CI/CD isolation and a new decision-model tag with auto-detection for GGUF models.

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

01 **Decision-model tag on Model pages** NEW 55

Adds a `decision-model` tag for model pages, shown with a dedicated icon and filterable under the Misc section of the Models page. GGUF decision models are tagged automatically from their llama.cpp metadata on upload, so no manual tagging is needed for that format.

— Explains mechanism and auto-detection but no direct action path

changelog-20261013-4ed10204

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

KoboldCpp

Feedback ⓘ v1.123 NOTES ↗ CODE ↗ RANK

KoboldCpp runs GGUF language models locally as a single-file application with a built-in web UI and API.

> KoboldCpp v1.123 overhauls its built-in Agent with approval modes, session persistence, and a dedicated GUI tab, extends batched inference to full sampler/grammar/vision support, and adds Qwen Image 2.1 generation alongside several CLI flag deprecations.

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

01 Approval modes and session persistence in KoboldCpp Agent

NEW 95

Adds agent commands `/confirm NAME on|off|auto` to set approval mode globally or per-tool (with `auto` auto-approving read/write within `workdir` but requiring confirmation for `shell` calls or writes outside `workdir`), `/save` and `/load` to persist and restore agent sessions, `/maxsteps` to configure max automated agent steps, `/autocompact N` to trigger context compaction at a token count, `/usage` to display token usage statistics, and `/paste` for multiline input. A new `--agentload [file]` CLI flag reloads a saved KoboldCpp Agent session directly at launch, and the Agent has moved to a dedicated Agent tab in the GUI launcher.

Resume a previously saved agent session at startup to continue a long-running coding or automation task without re-establishing context.

```
$ koboldcpp.exe --model mymodel.gguf --agent --agentload
mysession.json
```

— Names every command and flag with a runnable example v1.123

02 Full sampler, grammar, and vision support in batched mode

IMPROVED 75

`--parallelrequests N` now supports all standard samplers, grammar, tool calls, and vision in batched (parallel) mode, letting multiple parallel inference requests serve several API clients simultaneously with full feature parity.

Run multiple parallel inference requests with full sampler and grammar support to serve several API clients simultaneously with low latency.

```
$ ./koboldcpp-linux-x64 --model mymodel.gguf --parallelrequests 4
```

— Names the flag and includes a working example command v1.123

03 **`--jinjatools` and `--multiuser` flags deprecated** DEPRECATED

70

`--jinjatools` is deprecated in favor of `--jinja`, which now always enables Jinja tools automatically and falls back gracefully on failure; `--multiuser` is deprecated in favor of `--queuelimit` for setting the connection queue limit.

— Names exact deprecated and replacement flags v1.123

THINNER COVERAGE BELOW

04 **Qwen Image 2.1 model support** NEW

30

Adds support for the Qwen Image 2.1 model for image generation and editing.

— Thin mention with no usage or config detail v1.123

05 **Saved custom OpenAI endpoint favorites in Kobold Lite** NEW

30

Allows saving multiple custom OpenAI endpoints as favorites in Kobold Lite.

— Brief mention without configuration or UI detail v1.123

└──▷ **BREAKING ON UPGRADE**

! The `--jinjatools` flag is deprecated; use `--jinja` instead, which now automatically enables Jinja tools.

! The `--multiuser` flag is deprecated; use `--queuelimit` to set the connection queue limit.

WAS THIS USEFUL? 👍 👎

◆ AI / LLM Gateways

NVIDIA Personal AI Router

Feedback ⓘ v1.0.0 NOTES ↗

RANK

Personal AI Router virtually distributes LLM inference across a user's connected home devices.

> Personal AI Router's v1.0.0 release adds llama.cpp as a managed engine with Hugging Face GGUF model support, routes the Anthropic Messages API through Ollama and LM Studio, validates RTX Spark N1X hardware, and ships a redesigned terminal interface — alongside a breaking CORS policy change and a macOS app rename.

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

01 CORS enforcement with OLLAMA_ORIGINS engine launch options

90

BREAKING

PAIR's local endpoints now enforce each engine's own CORS policy instead of allowing every browser origin, so browser-based clients may stop working until the engine is configured to permit them. New engine launch options let you set environment variables and arguments for any PAIR-managed engine, local or paired, including `OLLAMA_ORIGINS`, to control settings such as context size and CORS origins.

Allow a specific browser-based client to reach a PAIR-managed Ollama engine after the CORS policy change locks down origins.

📍 In the PAIR engine settings for Ollama, add the environment variable: `OLLAMA_ORIGINS=https://my-browser-client.example.com`

— Names config var, exact fix, and before/after behaviour v1.0.0

02 macOS app renamed to NVIDIA PAIR

65

BREAKING

The macOS app is renamed from `PAIR.app` to `NVIDIA PAIR.app`; the new app installs alongside the old one and you must manually remove `PAIR.app` to avoid running two copies, though settings, cluster membership, and models carry over automatically.

— Gives exact app names and required manual action v1.0.0

THINNER COVERAGE BELOW

03 llama.cpp managed engine with Hugging Face model downloads

NEW

55

Adds llama.cpp as a managed engine: install, manage, and download compatible GGUF models from Hugging Face, then route inference requests across paired devices.

— Names engine and format but no exact command v1.0.0

04 **Anthropic Messages API routing via Ollama and LM Studio** NEW

55

Adds Anthropic Messages API routing through Ollama and LM Studio, enabling compatible clients such as Claude Code to use PAIR's model-aware routing and failover.

— Names API and clients but no setup steps given v1.0.0

05 **Redesigned terminal interface with Jobs tab** IMPROVED

55

Adds a `Jobs` tab to the terminal interface that tracks inference work across paired devices and runs the Inference Demo, giving headless machines the same core tools as the desktop app. Also redesigns the terminal interface layout around each device, consolidating engine management, model downloads, and device pairing in one place.

— Describes new tab and layout but no exact navigation steps v1.0.0

06 **RTX Spark N1X hardware support** NEW

45

Adds RTX Spark N1X hardware support: PAIR's Windows on ARM build is now validated with llama.cpp and CUDA on RTX Spark N1X.

— States validation scope but no usage instructions v1.0.0

└──▷ **BREAKING ON UPGRADE**

! PAIR's local endpoints now enforce each engine's own CORS policy instead of allowing every browser origin; browser-based clients may stop working until the engine is configured to permit them (e.g., set `OLLAMA_ORIGINS` in Ollama's launch options).

! The macOS app is renamed from `PAIR.app` to `NVIDIA PAIR.app`; the new app installs alongside the old one and you must manually remove `PAIR.app` to avoid running two copies (settings, cluster membership, and models carry over automatically).

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

evalscope

Feedback ⓘ v1.13.0 NOTES ↗

A streamlined and customizable framework for efficient large model (LLM, VLM, AIGC) evaluation and performance benchmarking.

Adds `eval_type='systemone_api'`, on-demand adapter loading, MMMU multi-image perf mode, per-turn request IDs, explicit SLA threshold units, and a version-display Dashboard.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.13.0
https://github.com/modelscope/evalscope.git
# already have the repo? check out this version:
$ git checkout v1.13.0
```

Adds `eval_type='systemone_api'` to support text single-choice decision models through the existing model interface, retaining option probabilities, confidence, and request metadata.

Adds `--rerun-review` CLI flag (and `rerun_review=True` API argument) to rescore saved predictions under a new evaluation version without rerunning inference.

Adds a real MMMU multi-image mode for multimodal performance testing, with correct prompt/image placeholder ordering.

Adds explicit `s` / `ms` threshold units to SLA auto-tuning, alongside metric-validity checks and improved constraint/optimization searches.

NEW DETECTION Adds conversation IDs and per-turn request IDs to multi-turn performance testing, plus SWE-Smith dataset replication support.

+2 more

NEW DETECTION Terminal-Bench 4.0 agent evaluation benchmark

NEW DETECTION Decision evaluation benchmarks: ANLI, BoolQ, BANKING77, ContractNLI, RewardBench v1

UPDATED DETECTION Unified Math-Verify mathematical scoring across 29 benchmarks

└──▷ BREAKING ON UPGRADE

! Evaluation versions have changed for 29 mathematical benchmarks, 22 NER benchmarks, AI2D, and IFBench — old reviews cannot be reused across evaluation

versions; use `rerun_review=True` / `--rerun-review` to rescore saved predictions.

! AI2D `v1.1` uses corrected per-question images and results are not directly comparable with `v1.0`.

! Terminal-Bench 4.0 results are not directly comparable with 2.0/2.1.

! Math-Verify uses ANTLR 4.13.2, conflicting with `ms-vlmeval==0.0.20` which requires ANTLR 4.11.1 — joint installation with the `vlmeval` / `all` extras conflicts; use separate environments.

WAS THIS USEFUL? 👍 👎

MLflow

Feedback ⓘ v3.17.1 NOTES ↗ RANK

MLflow tracks, evaluates, and monitors machine learning and LLM applications, covering experiment tracking, tracing, prompt management, and a model registry.

> MLflow 3.17.1 adds Decision API support to the LLM judge framework and introduces a new Databricks review queue management module.

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

01 **Decision API in make_judge()** NEW 40

make_judge() now supports the Decision API for evaluation workflows.

— Names the exact function but not mechanism or usage v3.17.1

02 **Databricks review queues module** NEW 40

A new `mlflow.genai.databricks.review_queues` module provides Databricks review queue management.

— Names the module path but no detail on its operations v3.17.1

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

Docker Sandboxes

Feedback ⓘ v0.48.0 NOTES ↗ RANK

Docker Sandboxes runs AI agents in isolated Docker environments with controlled filesystem, network, and tool access.

> Docker Sandboxes v0.48.0 overhauls the CLI with a unified `--format` output system and extensive breaking schema/flag changes, while adding cloud sandbox lifecycle controls (TTL, timeout, detached mode, template upload), SSH-based desktop integration for ChatGPT and Claude Desktop, model-provider secret management, and reasoning-level control for sandbox runs.

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

01 CLI structured output and flag overhaul

BREAKING 100

Introduces a unified `--format json|yaml|table|github-issue` output system across the CLI, with `--json` as an alias for `--format json`, and makes `--format`, `--json`, and `--quiet` / `-q` mutually exclusive on commands with shared output flags. `sbx settings ls` gains `--no-trunc` (full untruncated values), `--json`, and `--format yaml` (full record including source and default); `sbx diagnose` replaces `--output` / `-o` with `--format table|json|yaml|github-issue`; `sbx prune --dry-run` gains `--format json` and `--format yaml`; cloud `ttl --format yaml` now emits YAML instead of JSON; `sbx events --json` emits newline-delimited JSON; `sbx mcp auth` gains `--format text|json`. Flags are renamed for consistency: `--detached` to `--detach` / `-d` on `sbx run` and `sbx env run`; `--yes` to `-y` / `-f` on `sbx logout`; `--auto-approve` to `-y` / `-f` on `sbx env plan`, `sbx env create`, and `sbx env run`; `--quiet` to `-q` on `sbx env create` / `sbx env run` (with a new `--no-progress` to suppress progress while still printing the creation summary) and on `sbx create` (replaced by `--no-progress`); `--quiet` / `-q` on `sbx policy log` is removed with no replacement; `-d` to `--description` on `sbx template save`; and `sbx mcp add --skip_auth` to `--skip-auth`. Structured output schemas change: `sbx ls --json` emits a bare sandbox array (was `{sandboxes: [...]}`) with `created_at`, `last_used_at` as RFC 3339 timestamps and a new `detached` field; `sbx template ls` emits a bare image array (was `{images: [...]}`); `sbx mcp ls` (local) emits a bare server array with a `gateway` field per row (was `{gateway: {...}, servers: [...]}`); `sbx policy log` becomes `{updated_at, allowed: [...], blocked: [...]}` for both local and cloud, replacing the local `allowed_hosts` / `blocked_hosts` format and the camelCase cloud document; `sbx --cloud policy ls` fields change from camelCase to `snake_case`; `sbx ports` local rows use `host_ip`, `host_port`, `sandbox_port`, `protocol` and cloud rows use `sandbox_port` and `url` (replacing `port`), with publish/unpublish results now returning `action: published` / `action:`

`unpublished`; `sbx volume create`, `sbx volume ls`, and `sbx volume inspect` rename `ID`, `Name`, `CreatedAt` to `id`, `name`, `created_at`, and `sbx volume inspect` now defaults to human-readable output, requiring an explicit `--format` for structured data; and `sbx rm`, `sbx stop`, `sbx template rm`, and `sbx volume rm` now return an array of result objects with `action`, `resource`, `name`, and optional `id` / `error`. Unsupported `--format` values now fail at flag parsing instead of falling back to human-readable output.

Inspect the full setting record — including source and default — when troubleshooting an unexpected effective value.

```
$ sbx settings ls --format yaml
```

Feed all current settings into a script or pipeline for automated auditing.

```
$ sbx settings ls --json
```

Extract sandbox names from `sbx ls` in a CI pipeline after the v0.48.0 output shape change — the wrapper object is gone, so the jq path is now `.[].name`.

```
$ sbx ls --json | jq -r '.[].name'
```

Run a dry-run prune in YAML format to review what would be removed before committing — useful for scripted cleanup jobs that need machine-parseable output.

```
$ sbx prune --dry-run --format yaml
```

Stream sandbox events as newline-delimited JSON for ingestion by a log aggregator or SIEM.

```
$ sbx events --json
```

Parse sandbox names from structured output in CI — the array is now returned directly, so the jq filter changes from `.sandboxes[].name` to `.[].name`.

```
$ sbx ls --format json | jq '.[].name'
```

— Every renamed flag and schema change named verbatim with migration path. v0.48.0

02 **Cloud mode enhancements** NEW 90

Adds `sbx --cloud env create <file>` and `sbx --cloud env run <file>` subcommands to create and attach to cloud-hosted sandbox environments, with `--ttl` and `--on-timeout` (`restart`, `stop`, or `delete`) flags controlling sandbox expiration and timeout behaviour; `sbx move --to cloud` gains the same `--ttl` and `--on-timeout restart` flags (restart requires a TTL of at least one hour), extending always-on support beyond `sbx --cloud create`; `sbx --cloud run` gains `--detach` to start a cloud sandbox in the background without holding a terminal session; cloud mode rejects unsupported local settings (GPU, USB, display, shared skills, templates, governance profiles, clone options, MCP server definitions) before any host commands or provisioning run; and the new `sbx --cloud template load` subcommand uploads a local template tar file into the cloud template store, requiring `--cpus` and `--memory-mib` flags matching a supported cloud size, with `--capture-mode all` to capture memory and checkpoint state when uploading a local template.

Create a cloud sandbox with a 4-hour TTL that automatically restarts on expiry — useful for long-running unattended agent jobs.

```
$ sbx --cloud env create --ttl 4h --on-timeout restart
./cloud.sbxenv.yaml
```

Move a local project to the cloud with a 4-hour TTL and automatic restart on expiry, so the sandbox keeps running unattended without manual intervention.

```
$ sbx move local-project --to cloud --ttl 4h --on-timeout restart
```

Run a long-lived cloud sandbox in the background so your terminal is free while the task executes.

```
$ sbx --cloud run --detach claude --name cloud-task
```

Save a running cloud sandbox with full memory and checkpoint state so it resumes instantly instead of cold-booting.

```
$ sbx --cloud template save --capture-mode all
```

Upload a locally built template tar into the cloud store, sized to match a supported cloud instance, so it can be launched as a cloud sandbox.

```
$ sbx --cloud template load my-template.tar cloud-template --cpus 2 --memory-mib 4096
```

— Full flag set and restart precondition both given. Subcommand, required flags, and capture mode all named.

v0.48.0

03 **Reasoning-level control for model runs** NEW 85

Adds `--variant` flag to `sbx run` to select a reasoning level (`minimal` , `low` , `medium` , `high` , `xhigh` , `max` , `none` , `thinking`) supported by the model and agent, e.g. `sbx run --provider openai --model <MODEL_NAME> --variant high codex` .

Start Codex at high reasoning effort using an OpenAI model that supports it — useful when you want deeper chain-of-thought without changing the model.

```
$ sbx run --provider openai --model <MODEL_NAME> --variant high codex
```

— Flag, full enum of values, and runnable example given. product docs

04 **Model-provider API key secrets** IMPROVED 85

Adds `sbx secret set <PROVIDER_ID>` to store model-provider API keys as sandbox-scoped secrets, which take precedence over global secrets and avoid daemon restarts when keys change; supports an optional host environment variable as a fallback API key source for custom endpoints, and the host proxy automatically injects stored credentials into authentication headers on requests to the host's service, keeping real API keys outside the sandbox. Adds `--env-name` flag to `sbx secret ls` and `sbx secret rm` (replacing the removed `--env` flag), and allows sandbox-scoped custom secrets to override global secrets that share the same environment variable name.

Store an OpenAI API key as a sandbox-scoped secret so the host proxy can inject it automatically — no daemon restart needed when the key rotates.

```
$ sbx secret set openai
```

— Names commands, flags and precedence rules; mechanism explained. v0.48.0

05 **SSH setup for ChatGPT and Claude Desktop** NEW 80

Adds `sbx setup ssh --auto-create` on macOS and Windows to register dedicated Docker Sandbox SSH connections for ChatGPT and Claude Desktop, creating an isolated

sandbox for each app on first connection; `sbx setup ssh remove` tears down the managed SSH config and app connections without affecting running sandboxes.

Wire ChatGPT and Claude Desktop into their own dedicated sandboxes so each app gets an isolated environment on first connection.

```
$ sbx setup ssh --auto-create
```

Remove all managed SSH config and desktop app connections for ChatGPT and Claude Desktop when you no longer want the integration.

```
$ sbx setup ssh remove
```

Register a dedicated Codex and Claude Desktop sandbox via SSH on macOS or Windows — each AI desktop client gets its own persistent sandbox on first connect.

```
$ sbx setup ssh --auto-create
```

— Names both commands and their effect; no deeper mechanism given. v0.48.0

06 **Confirmation prompt and automation behaviour changes** BREAKING

80

At selection and text prompts, Enter now accepts the default and Ctrl-D cancels; declining a required confirmation returns a non-zero exit status consistently for removal, reset, and prune operations. Piping confirmation input (e.g. `echo y | sbx secret set …`) no longer confirms the operation — commands requiring confirmation now fail with exit status 1 when stdin is not a terminal. Read-only invocations and runs with `--force` no longer start an interactive sign-in flow; users must sign in before running automation that requires authentication. Without a terminal, `sbx diagnose --upload` saves the diagnostic bundle but skips uploading it and exits with status 0, and `sbx create` / `sbx run` fail if the workspace directory does not exist.

— Exact exit-code and prompt behaviour changes named; no single command example. v0.48.0

07 **Headless login with Docker PAT** NEW

75

Adds `sbx login --username <your-docker-id> --password-stdin` to authenticate the CLI in CI and headless environments using a Docker Personal Access Token (PAT) piped via stdin.

Authenticate the `sbx` CLI in a CI pipeline where no browser is available, using a Docker PAT stored as a secret.

```
$ echo "$DOCKER_PAT" | sbx login --username <your-docker-id> --password-stdin
```

— Exact command and use case given, no further mechanism. product docs

---

08 **Detached run mode and listing columns** NEW 75

Adds `sbx run --detach` flag to keep sandboxes running in detached mode, shown as (detached) in the `STATUS` column of sandbox listings; adds a `CREATED` column to listings alongside `STATUS`, `SANDBOX`, `AGENT`, `PORTS`, and `WORKSPACE`; both `sbx ls` and `sbx inspect` now identify detached sandboxes.

Run a Claude sandbox in the background and continue using your terminal while the sandbox stays alive.

```
$ sbx run --detach
```

— Flag and exact column names given; limited mechanism beyond display. v0.48.0

---

09 **Detach MCP server without restart** NEW 75

Adds `sbx mcp unload NAME --sandbox SANDBOX` to detach a loaded MCP server from a running sandbox without restarting the sandbox agent.

Detach a loaded MCP server from a running sandbox without restarting its agent — useful when rotating server configurations mid-session.

```
$ sbx mcp unload my-server --sandbox my-sandbox
```

— Exact command with required flags, minimal further detail. v0.48.0

10 **V3 kit capabilities and container environment** NEW

75

V3 kits gain support for capability groups, `kit.env` references, bundled agent skills, and the `ssh-agent@1` and `git-identity@1` capabilities; `sbx kit add` now accepts kits with startup commands that run after the kit is added and on each subsequent sandbox start; `sbx kit validate` now rejects kits whose credentials route the same host, identifying the conflicting services; and a new `SBX_AGENT` environment variable is exposed inside sandbox containers, giving the provisioned agent name for use in kit hooks and scripts.

— Named capabilities and validation rule, but no runnable example shown. Variable named with purpose, no usage example given.

v0.48.0

11 **Writable environment files in local sandboxes** NEW

70

Adds `sandboxOptions.writableEnvFiles: true` config key to allow the agent to edit shared environment files in local sandboxes; it is opt-in, and the sandbox plan reports the file as writable when enabled.

Allow a trusted agent to modify the shared environment file in a local sandbox so updated commands take effect on subsequent runs.

yaml

```yaml
sandboxOptions:
  writableEnvFiles: true
```

— Config key and behaviour named; no further mechanism detail. product docs

12 **SSH-agent forwarding for Windows OpenSSH and 1Password**

IMPROVED

70

Supports the `\\.\pipe\openssh-ssh-agent` named pipe on Windows for SSH-agent forwarding, compatible with Windows OpenSSH and 1Password's SSH agent, and falls back to these agents when `SSH_AUTH_SOCK` is not set.

— Exact pipe path and compatible agents named, no command shown. v0.48.0

13 **Unattended sandbox plan approval flag** NEW

60

Adds `sbx env create --force` flag to approve a sandbox plan for a single unattended invocation without recording persistent consent.

Approve and run a sandbox plan once in an unattended CI pipeline without storing persistent consent.

```
$ sbx env create --force
```

— Single flag named with clear effect, thin otherwise. product docs

└──▷ BREAKING ON UPGRADE

- ! `--detached` is renamed to `--detach` / `-d` on `sbx run` and `sbx env run`; scripts using `--detached` will break.
- ! `--yes` is renamed to `-y` or `-f` on `sbx logout`; scripts using `--yes` will break.
- ! `--auto-approve` is renamed to `-y` or `-f` on `sbx env create` and `sbx env run`; scripts using `--auto-approve` will break.
- ! `--quiet` is renamed to `-q` on `sbx env create`, `sbx env run`, and `sbx policy log`; scripts using `--quiet` on these commands will break.
- ! The `-d` shorthand for description on `sbx template save` is renamed to `--description`; scripts using `-d` for description will break.
- ! `--output` is renamed to `-o` / `--format` on `sbx diagnose`; scripts using `--output` will break.
- ! `sbx mcp add --skip_auth` is replaced by `--skip-auth`; scripts using `--skip_auth` will break.
- ! `sbx ls --json` now emits a bare array instead of `{sandboxes: [...]}`. JQ filters such as `.sandboxes[].name` must be updated to `.[].name`.
- ! `sbx template ls --json` now emits a bare image array instead of `{images: [...]}`; parsers expecting the wrapper object will break.
- ! `sbx mcp ls --json` (local) now emits a bare server array instead of `{gateway: {...}, servers: [...]}`. Cloud rows differ in structure from local registration rows.
- ! `sbx policy log` structured output shape changes to `{updated_at, allowed: [...], blocked: [...]}` for both local and cloud; parsers expecting the previous shape will break.
- ! `sbx --cloud policy ls` field names change from camelCase to `snake_case`; parsers using camelCase field names will break.
- ! `sbx ports` local rows rename fields to `host_ip`, `host_port`, `sandbox_port`, `protocol`; cloud rows replace `port` with `sandbox_port` and `url`; parsers using old field names will break.
- ! `sbx volume create`, `sbx volume ls`, `sbx volume inspect` rename `ID`, `Name`, `CreatedAt` to `id`, `name`, `created_at`; parsers using the old capitalised field names will break.
- ! `sbx volume inspect` now defaults to human-readable output; scripts relying on structured output from `sbx volume inspect` must add an explicit `--format json` or `--format yaml`.
- ! `sbx rm`, `sbx stop`, `sbx template rm`, and `sbx volume rm` now return an array of result objects with `action`, `resource`, `name`, and optional `id` and `error`; parsers expecting prior output shapes will break.
- ! `--format`, `--json`, and `--quiet` (`-q`) are mutually exclusive on commands with shared output flags; scripts passing more than one of these together will now fail.

- ! `--detached` on `sbx run` and `sbx env run` is rejected; use `--detach` or `-d`.
- ! `--yes` / `-y` on `sbx logout` is rejected; use `--force` or `-f`.
- ! `--auto-approve` / `-y` on `sbx env plan`, `sbx env create`, and `sbx env run` is rejected; use `--force` or `-f`.
- ! `--quiet` / `-q` on `sbx create` is rejected; use `--no-progress` (still prints the creation summary).
- ! `-d` for the description on `sbx template save` is rejected; use `--description`.
- ! `--output` / `-o` on `sbx diagnose` is rejected; use `--format table`, `--format json`, `--format yaml`, or `--format github-issue`.
- ! `--env` on `sbx secret ls` and `sbx secret rm` is rejected; use `--env-name`.
- ! `--quiet` / `-q` on `sbx policy log` is removed with no replacement.
- ! `--format`, `--json`, and `-q` are mutually exclusive; `--json --format json` is now rejected.
- ! Unsupported `--format` values now fail during flag parsing instead of falling back to human-readable output.
- ! `sbx ls --json` returns the sandbox array directly (e.g. `.[].name`) instead of `{"sandboxes": [...]}` (e.g. `.sandboxes[].name`).
- ! `sbx template ls --json` returns the image array directly instead of `{"images": [...]}`.
- ! `sbx mcp ls --json` locally returns the server array directly instead of `{"gateway": {...}, "servers": [...]}`.
- ! `sbx policy log` output changes to `{updated_at, allowed: [...], blocked: [...]}` with new entry fields, replacing the previous `allowed_hosts` / `blocked_hosts` local format and the cloud camelCase document.
- ! `sbx --cloud policy ls` field names change from camelCase to `snake_case`.
- ! `sbx ports` cloud rows use `sandbox_port` instead of `port`; local rows use `host_ip`, `host_port`, `sandbox_port`, and `protocol`.
- ! `sbx ports --publish` / `--unpublish` locally now emit result rows with `action: published` or `action: unpublished` instead of human-readable lines.
- ! Cloud `volume create`, `volume ls`, and `volume inspect` field names change from PascalCase (`ID`, `Name`, `CreatedAt`) to `snake_case` (`id`, `name`, `created_at`); `volume inspect` now defaults to a human-readable view — use `--format json` for JSON.
- ! `sbx rm`, `sbx stop`, `sbx template rm`, and `sbx volume rm` structured output now returns an array of result objects with `action`, `resource`, `name`, and optional `id` and `error`.
- ! Piping confirmation input (e.g. `echo y | sbx secret set …`) no longer confirms the operation; commands that require confirmation fail with exit status 1 when stdin is not a terminal.
- ! Declining a confirmation now returns a non-zero exit status consistently for removal, reset, and prune operations.
- ! Read-only invocations and runs with `--force` no longer start an interactive sign-in flow; sign in before running automation that requires authentication.
- ! Without a terminal, `sbx diagnose --upload` saves the diagnostic bundle but skips uploading it and exits with status 0 — a successful exit no longer guarantees an upload

occurred.

! Without a terminal, `sbx create` and `sbx run` fail if the workspace directory does not exist.

! Ctrl-D at a selection or free-text prompt now cancels instead of accepting the default.

WAS THIS USEFUL?  

AI MODELS

◆ Frontier Models

Google Gemini 4 Argon

Feedback ⓘ released · Oct 13, 2026

MODEL ↗ DOCS ↗ NOTES ↗ BLOG ↗

Google: "Announcing Gemini 4 Argon, our frontier model for real-world coding, enterprise knowledge work, and cyber defense, rolling out soon."

Gemini 4 Argon from Google was released on 30 Sep 2026. 1049K-token context; $2.00 in / $10.00 out per million tokens (OpenRouter).

Gemini API adds long decode on Gemini 4 Argon, enabling outputs past 64K tokens up to 1M via background execution on the Interactions API.

Enables `background: true` with `max_output_tokens` above 65,536 (up to 1,048,576) on client.interactions.create() to unlock long decode, generating responses in automatic 64K hops stored under a single interaction ID.

Exposes `interaction.steps` (filtered by `step.type == 'model_output'`) to read the full multi-hop response by joining all `model_output` step contents.

Exposes `interaction.usage` with fields `total_input_tokens`, `total_cached_tokens`, `total_output_tokens`, `total_thought_tokens`, and `total_tokens` to track token consumption across all hops.

+28 more

└─▷ **BREAKING ON UPGRADE**

- ! Setting `max_output_tokens` above 65,536 with `background: false` returns a 400 error ('max_output_tokens greater than 65536 is not yet supported with `background=false`').
- ! Background interactions require `store: true`; omitting it returns a 400 error ('`store=true` is required for background interactions').
- ! Explicit context caching (`cachedContents` / `caches.create`) is not supported on `gemini-4-argon`; calls that create or reference `cached_content` must be removed.
- ! Sampling parameters (not individually named in the source) are silently ignored by `gemini-4-argon` if provided.
- ! `gemini-4-argon` is not available on the Free tier or the Google AI Pro plan.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · October 13, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue