THE AI TOOLCHAIN

NO. 074

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED OCTOBER 26, 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 90 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 20 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.

EVALUATE

Watch what an agent actually typed in the sandbox when an eval fails

vim ? help

Inspect AI 0.3.285 adds a Terminals tab that replays recorded terminal sessions from an eval run, and consolidates the scattered info views into one Info tab; Braintrust's Loop can now read GitHub source during a trace investigation. Both close the same gap — a failing agent run tells you the output but not the command or the code path that produced it, so you no longer have to re-run locally and hope it reproduces.

Inspect AI · Braintrust

DEPLOY

Send a million-token prompt to a hosted model and dial how hard it thinks

Inference.net's MiniMax M3.1 Flash is serverless with a 1M-token context window and selectable reasoning effort, so whole-repo or long-document prompts run without standing up GPUs, and cheap calls don't pay for deliberation they don't need. CTranslate2 v4.8.4 adds experimental Drama 3 text-to-speech support plus the ops to run it, and Python 3.15 wheels.

Inference.net · CTranslate2

BUILD

Keep a coding agent's context across sessions and split a goal across a crew

Kiro adds cross-session Memory so an agent stops relearning project facts on every new task, and Kiro Crew 0.10.0 adds a team-lead agent that splits a goal into sub-goals with per-agent environment and memory scoping. Qwen Code's SDK gains session resume and export plus a concurrent batch runner, so long agent runs survive a restart and fan out instead of queuing.

Kiro · Kiro Crew

OFFENSIVE SECURITY

◆ Exploitation & C2

xalgorix

Feedback ⓘ v4.6.171 NOTES ↗ RANK

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

> xalgorix's latest releases focus on making long-running scans terminate cleanly — via two new convergence environment variables and an endpoint-hygiene pass — alongside hardened API key pooling and subprocess secret isolation.

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

01 Principled scan finish via `XALGORIX_CONVERGE_FINISH` NEW

93

Adds `XALGORIX_CONVERGE_FINISH` environment variable (default off, byte-identical behavior when off) that lets a saturated scan — foundational discovery done, no open single-endpoint hypotheses, and injection classes exercised past a breadth threshold — close as coverage-sufficient instead of holding the scan open forever. Remaining untested (route, class) pairs are recorded as residual in finish telemetry so the report shows what was and wasn't individually retested, with a stuck authentication lane closing as `completed-with-blocked-work` (resumable) rather than silent success; VCS metadata (`.git` / `.svn` / `.hg`), server-config and secret files (`.htaccess` / `.htpasswd` / `web.config`), logs, OS cruft, backups, and the synthetic 404-baseline probe are restricted to information-exposure checks only, removing their unsatisfiable injection-class obligations from the coverage matrix. A follow-up fix ensures scans using this flag now finalize correctly, closing pending model-generated specific-endpoint hypotheses as logged residual (endpoint, class) pairs in the report instead of blocking completion.

— Full mechanism, scope and bug-fix follow-up named; no runnable example given. v4.6.169 v4.6.171

02 Scan convergence via `XALGORIX_CONVERGE_RECONCILE` NEW

88

Adds `XALGORIX_CONVERGE_RECONCILE` environment variable (default off) that reconciles class-less exploratory tasks once all vuln-class, recon, dirbust, and auth tasks are terminal and the minimum-iteration floor has passed, stopping infinite task churn once a scan reaches 100% coverage.

Enable scan convergence so long-running scans stop churning once full endpoint × class coverage is achieved, rather than running indefinitely.

```
$ XALGORIX_CONVERGE_RECONCILE=true xalgorix scan <target>
```

— Exact env var, trigger conditions, and runnable command given. v4.6.166

03 **API key pool rotation strategies** IMPROVED 85

Adds `XALGORIX_KEY_POOL_STRATEGY` environment variable with two selectable pool strategies: `roundrobin` (default, both keys share load with automatic cooldown) and `sticky` (one key at a time, failover only on rate limit). Rotation now also fires when a request already carries any pooled key from `XALGORIX_API_KEYS`, so operators who leave `XALGORIX_API_KEY` empty no longer have all traffic pinned to a single key.

— Exact env vars and strategy values named; no example command shown. v4.6.170

04 **Endpoint inventory hygiene via `XALGORIX_ENDPOINT_HYGIENE`**

NEW 78

Adds `XALGORIX_ENDPOINT_HYGIENE` environment variable (default off) enabling endpoint validation at every recording point plus a self-heal pass that scrubs CLI tool-config paths, unexpanded payload/shell template variables, and out-of-band callback URLs from the endpoint inventory, frozen coverage baseline, and per-endpoint × class coverage map on each plan reconcile.

— Env var and scrubbed artifact types named; no usage example given. v4.6.168

05 **Secret isolation hardening for subprocess tools** IMPROVED 68

Introduces a shared `tools.SecretEnvKey` deny-list backed by both the terminal and Python tools so operator secrets (integration tokens, provider keys) are stripped from child-process environments consistently, closing a gap where non-namespaced secrets were previously visible to `terminal_execute` subprocesses.

— Names exact mechanism and prior gap; no reader-facing action. v4.6.166

WAS THIS USEFUL? 👍 👎

BUILD

◆ AI Coding Agents

Amazon Kiro

Feedback ⓘ changelog · Oct 26, 2026 CHANGELOG ↗

RANK

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

> Kiro's single release this window adds cross-session Memory, per-workspace trust prompts, V3 model-fallback and workflow controls, and ships Crew 0.9.0 with profile cards, a Dynamic Dashboard preview and remote crew session windows — alongside several breaking defaults changes to verbosity, panels, and settings layout.

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

01 **Per-workspace trust prompts for cloned repos** NEW 78

Kiro CLI now stores a `workspaceTrusted` flag per workspace under `~/.kiro/workspace-roots/` and prompts 'Trust this workspace?' before loading hooks, agents, MCP servers, steering, workflows, or language server config from a cloned repo. Trust can be revoked by removing the flag from the stored workspace-root file.

Trust a freshly cloned repo so Kiro CLI loads its hooks and MCP servers — do this once per workspace root.

```
$ # Kiro CLI prompts on first interactive V3 launch in the
folder:
# 'Trust this workspace? Trust, continue / Don't trust, exit'
# To revoke trust later, remove 'workspaceTrusted' from the file
shown:
cat ~/.kiro/workspace-roots/<workspace-hash>
```

— Names exact storage path, flag, and prompt flow changelog-20261026-6ad8c3e7

02 **Cross-session Memory in Kiro IDE** NEW 77

Introduces Memory, which carries context such as repository conventions and prior decisions across chat sessions, controlled by a `Kiro Agent: Memory` setting that includes a Use only mode, with a `/memories` path for viewing and deleting stored memories. The Memory setting's default has changed to Off, so existing sessions that relied on it being on will stop accumulating memories until re-enabled.

— Explains mechanism and a default-off behavior change changelog-20261026-6ad8c3e7

03 **Dynamic Dashboard feature preview for crewmates** NEW

70

Adds `Settings > Developer > Feature Previews > Dynamic Dashboard`, giving each crewmate a Dashboard tab with 12 built-in project-report pages and on-demand pages generated in chat.

— Exact settings path and concrete page count changelog-20261026-6ad8c3e7

---

04 **Remote crew sessions feature preview** NEW 70

Adds `Settings > Developer > Feature Previews > Remote crew sessions`, opening a peer crew's session as a live window showing its transcript, running state, and approvals without storing data locally.

— Exact settings path and described live-view scope changelog-20261026-6ad8c3e7

---

05 **Model fallback command for V3 sessions** NEW 65

Adds a `/model fallback` command and `/model` settings pane in V3 sessions to save and persist model fallback configuration across sessions.

Set a model fallback in the current V3 session so work continues if the primary model is unavailable.

```
$ /model fallback
```

— Exact command given, mechanism thinly described changelog-20261026-6ad8c3e7

---

06 **Telegram Bot API reverse-proxy routing** NEW 60

Adds `TELEGRAM_API_BASE_URL` environment variable to route the Telegram Bot API through a reverse proxy when the public host is blocked.

Route the Telegram Bot API through a corporate reverse proxy so Kiro Crew can receive messages where the public Telegram host is blocked.

```
$ TELEGRAM_API_BASE_URL=https://telegram-proxy.internal kiro
```

— Named env var with a runnable usage example changelog-20261026-6ad8c3e7

THINNER COVERAGE BELOW

---

07 **Response verbosity defaults changed** BREAKING 55

New installs default `Settings > Chat > Transcript > Response Verbosity` to `Answer only`, while existing configs keep their stored level; the `Ultra` verbosity level is retired and any config that stored it now behaves as `Answer only`.

— Exact setting path and before/after default stated changelog-20261026-6ad8c3e7

08 **Crew 0.9.0 profile cards and status pills** NEW 50

Crew 0.9.0 adds per-crewmate Profile cards showing permissions, model, reasoning effort, schedules, and sessions, plus at-a-glance status pills.

— Names card contents but no path or command changelog-20261026-6ad8c3e7

09 **Automatic Slack threading for crew sessions** NEW 50

Adds `Settings > Messaging Channels > Slack > Connect new sessions to Slack automatically` to thread persistent dashboard sessions into Slack DMs with the bot.

— Exact settings path, limited mechanism detail changelog-20261026-6ad8c3e7

10 **Live markdown styling in chat composer** NEW 50

Adds `Settings > Chat > Composer > Style Markdown While Typing` option to render bold, italic, strikethrough, and inline code as you type.

— Exact settings path and listed formatting types changelog-20261026-6ad8c3e7

11 **Translucent panels default off** BREAKING 50

Translucent panels now default to off; turn on `Settings > Display > View > Translucent panels` to restore frosted-glass rendering.

— Exact setting path to restore prior behavior changelog-20261026-6ad8c3e7

12 **Agent settings renamed and reorganized** BREAKING 50

Agent Capabilities is relabeled Customize; Subagent Settings and Warm Pool have moved to `Settings > Agent Harness`, and Warm Pool is now labeled Pre-started Sessions.

— Names every relabeled and relocated setting changelog-20261026-6ad8c3e7

13 **Indemnity exception narrowed for excluded models** BREAKING 45

The Kiro Excluded Models indemnity exception no longer applies to Claude Fable 5.1, Claude Sonnet 5.5, GPT-5.6 Terra, and GPT-5.6 Luna.

— Names affected models but no action path given changelog-20261026-6ad8c3e7

14 **Retired turn relay makes old chats read-only** BREAKING 30

Old relay chats that ran on a peer crew through the retired turn relay now open read-only; sending to them is no longer possible.

— States impact but no named config or workaround changelog-20261026-6ad8c3e7

└──▷ ALSO FROM THESE RELEASES





└──▷ BREAKING ON UPGRADE

! New installs default `Settings > Chat > Transcript > Response Verbosity` to `Answer only`; existing configs keep their stored level.

! The `Ultra` verbosity level is retired — any config that stored `Ultra` now behaves as `Answer only`.

- ! Translucent panels now default to off; turn on `Settings > Display > View > Translucent panels` to restore frosted-glass rendering.
- ! Old relay chats that ran on a peer crew through the retired turn relay open read-only; sending to them is no longer possible.
- ! `Agent Capabilities` is relabeled `Customize`; `Subagent Settings` and `Warm Pool` have moved to `Settings > Agent Harness`, and `Warm Pool` is now labeled `Pre-started Sessions`.
- ! The `Kiro Excluded Models` indemnity exception no longer applies to Claude Fable 5.1, Claude Sonnet 5.5, GPT-5.6 Terra, and GPT-5.6 Luna.
- ! Memory setting default changed to Off — existing sessions that relied on Memory being on by default will no longer accumulate memories until the setting is re-enabled.

WAS THIS USEFUL? 👍 👎

Cognition Devin Desktop

Feedback ⓘ

changelog · Oct 26, 2026 | CHANGELOG ↗ | RANK

Devin Desktop is an agent-native code editor for developing software locally, remotely, and in containerized environments.

> Devin Desktop adds a unified Agent Window with side-by-side multi-session tiling, full Dev Container lifecycle support with rebuild commands and platform-scoped plugin manifests, a wave of new keyboard shortcuts, and several breaking changes to how agent edits are applied, saved, and queued.

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

01 Dev Container lifecycle support and rebuild commands NEW

95

Adds the `remote.devinDevContainers.dotfiles.*` config key and Dev Container lifecycle hooks (`onCreateCommand` , `updateContentCommand` , `postCreateCommand` , `postStartCommand` , `postAttachCommand`) read from `devcontainer.json` , plus support for Features, image metadata, `remoteEnv` , `customizations.vscode.settings` , extensions, `forwardPorts` , and `shutdownAction` . New `Dev Containers: Rebuild Container` and Rebuild Container Without Cache commands prompt a rebuild when `devcontainer.json` , the Dockerfile, or the Compose file changes, and a Reopen in Container prompt appears when opening a folder with Dev Container configuration.

— Every lifecycle hook, config key and command named verbatim changelog-20261026-6d073d67

02 Agent edit application now bypasses diff zones and save actions by default

90

BREAKING

Agent diff zones in Devin Settings → Editor → User Interface → Agent diff zones is now off by default, so agent edits apply directly to files without showing diff zones. The Auto-open edited files toggle in Devin Settings → Editor → User Interface now defaults to not opening files in the background when Devin edits them, and the new Run save actions on agent edits setting controls whether format-on-save and code actions run when the agent saves files (also off by default).

— Exact setting paths and new defaults are named changelog-20261026-6d073d67

03 New keyboard shortcuts for agent navigation and editing NEW

80

Adds `Cmd/Ctrl+.` to cycle agent mode, `Cmd/Ctrl+[` / `Cmd/Ctrl+]` to navigate back and forward, `Cmd/Ctrl+Shift+L` to copy a session link, `Cmd/Ctrl+Shift+E` to mark a session done, `Cmd/Ctrl+O` to open a local file in the Agent Window, `Cmd/Ctrl+Shift+F` to search the text of a session's files, and `Cmd/Ctrl+L` to quote selected editor code directly into the chat composer.

– All shortcut key combinations listed verbatim changelog-20261026-6d073d67

04 **Multi-session side-by-side Agent Window** NEW 75

The Agent Window now supports viewing multiple sessions side by side so users can monitor parallel workstreams at a glance; dragging a session card from the sidebar onto an already-open session panel tiles it, repeatable up to 8 active sessions.

View up to 8 active sessions side by side to monitor parallel workstreams at a glance.

📍 In the Agent Window, drag a session card from the sidebar onto an already-open session panel. Repeat up to 8 times to tile sessions.

– Exact UI drag steps and the 8-session limit are given changelog-20261026-6d073d67

05 **Platform-scoped plugin manifests** NEW 65

Plugin manifests can declare `platforms: [cloud]` or `platforms: [local]` so Devin Desktop only activates a plugin for the matching session type, e.g. scoping a plugin to cloud-only sessions so it never activates in Devin Local.

Scope a plugin to cloud-only sessions so it is never activated in Devin Local.

```yaml
platforms: [cloud]
```

– Exact manifest syntax with a usage example changelog-20261026-6d073d67

THINNER COVERAGE BELOW

06 **Enter/Cmd+Enter message handling changed during active turns** 55

BREAKING

`Cmd/Ctrl+Enter` now queues a message until the current turn ends, whereas previously it sent immediately; plain `Enter` now steers the currently running turn instead.

– Exact before/after keybinding behavior described changelog-20261026-6d073d67

07 **Appearance settings for interface and code fonts** NEW 40

`Settings → Preferences → Appearance` lets users pick interface and code fonts and their sizes.

— Only the settings path is given, no further mechanism `changelog-20261026-6d073d67`

---

08 **Worktree auto-deletion threshold removed** **BREAKING** 35

Worktrees are no longer deleted automatically once a repository has more than 19 of them, changing prior automatic cleanup behavior.

— States the threshold but no replacement mechanism given `changelog-20261026-6d073d67`

└──▷ ALSO FROM THESE RELEASES

.
.
.

└──▷ BREAKING ON UPGRADE

! Agent diff zones are off by default — agent edits now apply directly to files without showing diff zones. Re-enable in `Devin Settings → Editor → User Interface → Agent diff zones`.

! Devin no longer opens files in the background when it edits them. Re-enable via `Auto-open edited files` in `Devin Settings → Editor → User Interface`.

! Agent edits are saved without running format-on-save or code actions. Enable `Run save actions on agent edits` to restore the previous behavior.

! Worktrees are no longer deleted automatically once a repository has more than 19 of them.

! `Cmd/Ctrl+Enter` now queues a message until the current turn ends (previously sent immediately); `Enter` steers the running turn.

WAS THIS USEFUL?  

yetone magpie

Feedback ⓘ v0.1.1584 CODE ↗ RANK

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

> Magpie's v0.1.1584 release refines Kiro Crew's harness connection logic, auto-connecting and disconnecting based on the active harness.

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

01 **Kiro Crew harness auto-connect and fallback** IMPROVED

57

Kiro Crew now auto-connects the harness it is already set to, and disconnects it again when Kiro Crew is turned off. On a harness magpie does not wire directly (KAS, kiro-cli, deepseek), Kiro Crew falls back to the first harness connected to magpie (Codex, Claude Code, OpenCode, Pi, Goose) whose adapter is present.

— Describes behaviour and fallback order but no config/flag to act on. v0.1.1584

WAS THIS USEFUL? 👍 👎

Cline

Feedback ⓘ desktop-v0.0.49 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 CLI release adds new `cline hub status` and `cline doctor` diagnostic commands, embeds plugins and dashboard assets directly in compiled binaries, and ships safety guardrails for `run_commands` alongside expanded model lists and several reliability fixes to retries, auto-compaction, and the Gemini provider.

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

01 Hub status and doctor diagnostic commands NEW

75

Adds `cline hub status` and `cline doctor` commands with a 3-second health-check timeout so they no longer hang on a frozen Hub.

Check whether the Cline Hub is responsive — the command now times out after 3 seconds instead of hanging indefinitely.

```
$ cline hub status
```

Run the full diagnostic check without risk of hanging on a frozen Hub process.

```
$ cline doctor
```

— Exact runnable commands with stated timeout behaviour cli-v3.0.74

02 run_commands safety checks and PowerShell guidance IMPROVED

60

`run_commands` with an empty command list now returns an error instead of silently succeeding, and on PowerShell the tool explains quote-escaping rules and prompts the model to confirm recursive deletes before proceeding.

— Names the tool and exact behaviour changes cli-v3.0.74

THINNER COVERAGE BELOW

03 Expanded ClinePass and free model lists

UPDATED SUPPORTED PLATFORM NEW

55

ClinePass defaults to `Kimi K3.1 Pro` and adds `MiniMax M3.1 Flash` and `DeepSeek V4.1 Flash 1017` to its model list, while the Cline free model list adds

Reflection Beam, `Nemotron 3.5 Lightning`, and `Laguna S 2.1`.

— Names every model added but no usage mechanism cli-v3.0.74

04 **Gemini provider stops sending native search** IMPROVED

55

Gemini API keys via the `gemini` provider no longer send native Google Search, preventing quota 429 errors on free-tier keys and tool-drop bugs on Gemini 2.x.

— Names provider, error code and affected model line cli-v3.0.74

05 **Compiled binary embeds plugins and dashboard** IMPROVED

45

Compiled CLI binaries now load plugins and serve the dashboard from resources embedded in the binary, removing external asset dependencies.

— Explains mechanism but no flag or path to act on cli-v3.0.74

06 **Model request retry cap reduced to 3** IMPROVED 35

Model request retries are capped at 3 attempts (down from 5).

— States exact before/after numbers only cli-v3.0.74

07 **Kimi K3 max reasoning for high effort settings** IMPROVED

35

Kimi K3 now receives `max` reasoning for Extra High and Max effort settings.

— Names model and setting but little mechanism cli-v3.0.74

08 **Auto-compaction trigger fixes** IMPROVED 35

Auto-compaction no longer triggers for models with an unknown context limit, and no longer re-runs every few tool calls after the first compaction.

— Describes two behaviour fixes without further mechanism cli-v3.0.74

09 **Plugin slash commands run in active session** IMPROVED

30

Plugin slash commands now run in the active session instead of a detached context.

— Brief before/after with no further detail cli-v3.0.74

NEW INTEGRATION Liquid Inference and Munito AI model providers added

RETIRED INTEGRATION CrofAI, iFlow and Llama providers removed from catalog

NEW INTEGRATION Liquid Inference, Munito AI, and Munito AI (EU) providers added

RETIRED INTEGRATION CrofAI, iFlow, and Llama providers removed

desktop-v0.0.49 NOTES ↗ CODE ↗ LATEST STABLE

Cline desktop v0.0.49 adds a Changes pane, project-files panel, integrated terminal (Ctrl+backtick), configurable keybindings, and a Local Models settings page.

└──▷ GET THIS VERSION

```
$ git clone --branch desktop-v0.0.49
https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout desktop-v0.0.49
```

Adds an integrated terminal panel at the bottom of the window, toggled with `Ctrl+backtick`.

Adds a `Changes` button in the session header that shows line counts, lists files changed per turn or across the whole session, and displays a diff of each.

Adds a `project files panel` (folder button in the session header) with a file tree, read-only tabbed viewer, git status letters, and a File/Diff toggle for agent-edited files; works on local and SSH environments.

Adds a `Local Models` settings page for managing Ollama and LM Studio models, including setup guidance, model details and storage, and a composer chip showing loaded-model memory usage.

Adds configurable `Keybindings` in Settings (also reachable from the app and Help menus), with an option to require holding the shortcut to Quit — defaulting to 1 second on macOS.

+9 more

sdk/sdk/v0.2.0 NOTES ↗ CODE ↗ STABLE

SDK v0.2.0 adds session reattachment, hub-based capability routing, run-stall telemetry via `RunHealthObserver`, and `CLINE_PLUGIN_RUNTIME_DIR` support.

└──▷ GET THIS VERSION

```
$ git clone --branch sdk/sdk/v0.2.0
https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout sdk/sdk/v0.2.0
```

Reattaches to a live session on `session.start` instead of replacing its runtime; a start that includes a prompt while a run is in progress is rejected.

> Hub records every client that registered a capability for a session and routes each request to the first still-connected one, failing fast with an actionable message when none remain.
>
> Adds `agent.run-finished` and `agent.run-failed` telemetry events carrying stall signals from a new `RunHealthObserver`.
>
> Adds `CLINE_PLUGIN_RUNTIME_DIR` environment variable, resolved to an absolute path, to configure the plugin runtime directory.
>
> Adds `session.list` reporting of `activeCloudCommands` so callers can observe running cloud commands before the hub is replaced.
>
> +7 more

WAS THIS USEFUL? 👍 👎

Alibaba Qwen Code

Feedback ⓘ sdk-typescript-v0.1.23

NOTES ↗ RANK

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

> Qwen Code's TypeScript SDK gained multi-modal input, session resume and export, a concurrent batch runner, experimental LSP support and a Zed extension, alongside breaking changes that rename negative settings flags and remove the `read_many_files` and Smart Edit tools.

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

01 settings.env and contextWindowSize config keys NEW

75

Adds the `settings.env` field for configuring environment variables directly in settings, letting users set variables for a session without exporting them to a shell profile, and adds a new `contextWindowSize` configuration key.

Set environment variables for a session without exporting them to your shell profile.

json

```
{
  "settings": {
    "env": {
      "MY_API_KEY": "sk-...",
      "LOG_LEVEL": "debug"
    }
  }
}
```

— Exact config key with runnable JSON example sdk-typescript-v0.1.23

02 Removed tools: read_many_files and Smart Edit BREAKING

65

The `read_many_files` tool is removed (callers must use the `readManyFiles` utility via user `@ -commands` instead), and the Smart Edit tool is also removed, breaking workflows depending on it.

— Names old and replacement surface precisely No replacement or migration path given

sdk-typescript-v0.1.23

03 Session resume and auth options in SDK NEW

60

Adds `resume` and `continue` options and extended `authType` support to the SDK, plus support for resuming sessions with a session ID via the `query` command.

— Names flags but no usage example given sdk-typescript-v0.1.23

THINNER COVERAGE BELOW

04 **Settings flags renamed disable* to enable*** BREAKING

55

Negative settings flags are renamed to positive naming (`disable*` → `enable*`); any config referencing old `disable*` keys will stop working.

— Clear migration note but no full key list sdk-typescript-v0.1.23

05 **FORK_MODE for ProcessTransport** NEW

50

Adds `FORK_MODE` support to ProcessTransport for Electron IPC integration.

— Named flag and integration target, no usage shown sdk-typescript-v0.1.23

06 **Export command for session history** NEW

50

Adds an export command for session history in markdown and HTML formats.

— Formats named but command syntax absent sdk-typescript-v0.1.23

07 **Multi-modal input across content generators** NEW

45

Adds multi-modal input support (image, PDF, audio) across all content generators.

— Lists input types but no invocation shown sdk-typescript-v0.1.23

08 **extra_body support for OpenAI-compatible providers** NEW

35

Adds `extra_body` support for OpenAI-compatible providers.

— Named field only, no mechanism detail sdk-typescript-v0.1.23

09 **Concurrent runner for batch CLI execution** NEW

35

Adds a concurrent runner for batch CLI execution.

— States capability without mechanism or syntax sdk-typescript-v0.1.23

10 **Zed extension for agent server** NEW

35

Adds a Zed extension for the Qwen Code agent server.

— Names editor integration, no install steps sdk-typescript-v0.1.23

11 **Experimental LSP support** NEW

25

Adds experimental LSP support for code intelligence.

— Bare mention, no scope or setup described `sdk-typescript-v0.1.23`

└──▷ BREAKING ON UPGRADE

! Negative settings flags renamed to positive naming (`disable*` → `enable*`); any config referencing old `disable*` keys will stop working.

! The `read_many_files` tool is removed; callers must use the `readManyFiles` utility via user `@ -commands` instead.

! Smart Edit tool is removed; workflows depending on it will break.

WAS THIS USEFUL?  

Amazon Kiro Crew

Feedback ⓘ v0.10.0 NOTES ↗ RANK

Kiro Crew is an open-source workspace for coordinating AI coding agents.

> Kiro Crew's v0.10.0 release adds a goal-splitting team-lead agent, per-agent environment and memory controls, AWS Lambda MicroVM-backed crews, and a round of Chat settings for turn limits and transcript display.

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

01 Memory controls for writes, extraction and eviction NEW

90

Kiro Crew adds `memory.writes_enabled` and `memory.lesson_extraction_enabled` config keys to disable memory writes and lesson extraction independently, `memory.embedding_idle_evict_secs` to unload the embedding model after a configurable idle period (default one hour), and a `kirocrew memory forget` subcommand to remove a specific memory.

Prevent the embedding model from staying resident after an idle period to reduce memory on long-running hosts.

yaml

```yaml
memory:
  writes_enabled: true
  lesson_extraction_enabled: true
  embedding_idle_evict_secs: 3600
```

Remove a stale lesson from Kiro Crew's memory so it stops influencing future sessions.

```
$ kirocrew memory forget <memory-id>
```

— Four named config/command surfaces with runnable examples v0.10.0

02 Per-agent environment and context window config NEW

70

`agent.env` sets variables such as `HTTPS_PROXY` or `NODE_EXTRA_CA_CERTS` for every agent process, and `agent.model_context_tokens` declares the context window a model is served with.

Route every agent process through a corporate proxy and trust a private CA — set once, applies to all agents.

```yaml
agent:
  env:
    HTTPS_PROXY: "https://proxy.corp.example.com:8080"
    NODE_EXTRA_CA_CERTS: "/etc/ssl/certs/corp-ca.pem"
```

— Two named config keys with a runnable example v0.10.0

03 **Lambda MicroVM crews** NEW 70

A `microvm` block in `cloud.json` runs a crew in an AWS Lambda MicroVM billed per second, with a one-hour default lifetime.

— Named config block with billing and lifetime detail, no example v0.10.0

04 **Chat settings for turn limits and transcript display** NEW 60

Settings → Chat → Advanced → Turn Time Limit (seconds) caps how long a turn may run, with a per-session override via the timeout card; Settings → Chat → Transcript → Tool calls start expanded controls whether tool rows open by default; and Settings → Chat → Sessions → Show Full Titles While Scrolling keeps long session titles readable.

— Exact UI navigation paths named for each setting v0.10.0

THINNER COVERAGE BELOW

05 **Team-lead agent for goal splitting** NEW 50

`kirocrew-team-lead` is a new default agent that splits a goal into work items, opens a session per item, and handles small items itself.

— Behavior described but no invocation command shown v0.10.0

06 **Backend-per-task spawn_run parameter** NEW 50

A new `spawn_run` parameter lets each task pick its own backend, and the Subagents card lists every tool call a sub-agent makes.

— Parameter and UI card named, no usage example v0.10.0

07 **Multi-language Amazon Transcribe detection** NEW 50

`stt.transcribe_languages` lets Amazon Transcribe detect two to five languages in a single stream.

— Single config key with numeric range, no example v0.10.0

08 **AWS profile mapping per project** NEW 50

`aws.project_profiles` starts a session with the AWS profile mapped to its project folder.

— Single config key named, no example given `v0.10.0`

09 **GitHub Enterprise host registration** NEW 35

`dashboard.github_hosts` registers GitHub Enterprise hosts for the dashboard.

— Bare config key with minimal description `v0.10.0`

WAS THIS USEFUL? 👍 👎

AGENT

◆ MCP Servers & Tooling

Prefect FastMCP

Feedback ⓘ v4.1.4 NOTES ↗ RANK

FastMCP is a Python framework for building MCP servers and clients, handling transport, schema generation, authentication, and server composition.

> FastMCP v4.1.4 adds rotating Redis credential support for background tasks, exposes OAuth identity assertions, and lets clients tune SSE event sizing, alongside a breaking change to the keyring dependency.

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

01 **Redis credential providers in TasksExtension** NEW 70

`TasksExtension` now supports redis-py credential providers, supplying fresh credentials for each new Redis connection. Streaming providers (e.g. redis-entraid) are explicitly rejected until safe refresh is supported.

— Names mechanism and explicit limitation, but no config example v4.1.4

02 **KeyringStore now requires explicit dependency** BREAKING 70

Custom `KeyringStore` users must now explicitly install `py-key-value-aio[keyring]`; the `keyring` dependency is no longer bundled automatically.

— Gives exact install command needed to migrate v4.1.4

03 **max_sse_event_size on StreamableHttpTransport** NEW 65

`max_sse_event_size` is now forwarded through `StreamableHttpTransport`, enabling clients to control the maximum SSE event size.

— Names exact flag and transport class but no usage example v4.1.4

THINNER COVERAGE BELOW

04 **Identity assertions on OAuth provider wrappers** NEW 30

OAuth provider wrappers now expose identity assertions, making assertion data available to downstream handlers.

— States capability only, no surface names or usage path v4.1.4

└──▷ BREAKING ON UPGRADE

! Custom `KeyringStore` users must now explicitly install `py-key-value-aio[keyring]`; the `keyring` dependency is no longer included automatically.

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Inference Providers & GPU Clouds

Baseten

Feedback ⓘ changelog · Oct 26, 2026 CHANGELOG ↗

Baseten deploys and serves machine learning models as autoscaling inference endpoints, with a model API catalog and dedicated GPU deployments.

Adds GPU temperature/utilization metrics for Dedicated Inference; authentication errors shift to 401; unused API keys auto-revoked after 90 days.

Adds GPU temperature and utilization metrics in the Metrics tab for Dedicated Inference deployments.

Changes API authentication error responses from `403 Forbidden` to `401 Unauthorized` for missing, invalid, or expired credentials starting October 28, 2026.

Automatically revokes API keys that have not authenticated a request in 90 days, starting November 5, 2026.

NEW INTEGRATION elephant-1 and mammoth-1 model APIs

└──▷ BREAKING ON UPGRADE

! Starting October 28, 2026, Baseten APIs return `401 Unauthorized` instead of `403 Forbidden` for requests with missing, invalid, or expired credentials — any client logic branching on `403` for auth errors will break.

! Starting November 5, 2026, API keys unused for 90 days are automatically revoked — integrations relying on long-dormant keys will stop authenticating.

WAS THIS USEFUL? 👍 👎

Nebius AI Cloud

Feedback ⓘ October 19–25, 2026 (2026-10-26)

BLOG ↗ RANK

Nebius AI Cloud provides GPU compute, managed Kubernetes, distributed training infrastructure, and serverless AI workloads.

> Nebius AI Cloud introduced Devlab for infrastructure-free JupyterLab environments, made spot GPU pricing public with historical data, and added restart controls for Serverless AI endpoints, while retiring standalone applications and Container VMs in favor of Kubernetes and Devlabs.

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

01 **Public spot GPU pricing with history** NEW 75

Spot prices for preemptible GPU platforms are now publicly viewable without an account via the Nebius website or API, or with an account via the web console or CLI, with up to 30 days of price history to inform region selection or max-price policies.

– Specifies access channels, data window, and intended use

October 19–25, 2026 (2026-10-26)

02 **Serverless AI endpoint restart controls** NEW 60

Serverless AI endpoints can now be restarted via the web console, CLI, or API; restarted endpoints retain their public HTTPS URL and IP address.

– Names surfaces and behavior but no exact command syntax

October 19–25, 2026 (2026-10-26)

03 **Standalone apps and Container VMs retired** BREAKING 60

Standalone applications and Container VMs are no longer available; workloads such as JupyterLab, ComfyUI, and Qdrant must be migrated to Managed Service for Kubernetes clusters or Serverless AI Devlabs.

– Names affected workloads and the two migration targets October 19–25, 2026 (2026-10-26)

THINNER COVERAGE BELOW

04 **Support ticket priority tiers change** BREAKING 50

Starting November 2, 2026, support ticket priorities change to Medium (4-hour first response) and Low (1 business day); customers without a committed volume contract signed by November 1, 2026 lose current priority tiers and response times.

– Gives exact dates and response windows but no remediation link

October 19–25, 2026 (2026-10-26)

05 **Devlab managed interactive environments** NEW 45

Devlab is a new managed environment type that deploys JupyterLab or other interactive environments without requiring the user to manage underlying infrastructure. It also serves as a migration target for workloads previously run as standalone applications or Container VMs.

— Names the feature and purpose but no setup steps or flags given

October 19–25, 2026 (2026-10-26)

06 **Kubernetes SLA exclusions for EOL clusters** BREAKING 45

Starting November 2, 2026, the Managed Service for Kubernetes SLA does not apply to clusters running an end-of-life Kubernetes version or to unavailability caused by modifying preinstalled add-ons outside the public API.

— States exact exclusion conditions but no upgrade guidance

October 19–25, 2026 (2026-10-26)

07 **Token Factory fees now post-paid** IMPROVED 20

Token Factory fees have moved to a post-paid billing model.

— Single-sentence billing change with no further mechanism detail

October 19–25, 2026 (2026-10-26)

└──▷ **BREAKING ON UPGRADE**

! Standalone applications and Container VMs are no longer available; workloads (JupyterLab, ComfyUI, Qdrant, etc.) must be migrated to Managed Service for Kubernetes clusters or Serverless AI Devlabs.

! Starting November 2, 2026, support ticket priorities change to Medium (4-hour first response) and Low (1 business day); customers without a committed volume contract signed by November 1, 2026 lose current priority tiers and response times.

! Starting November 2, 2026, the Managed Service for Kubernetes SLA does not apply to clusters running an end-of-life Kubernetes version or to unavailability caused by modifying preinstalled add-ons outside the public API.

WAS THIS USEFUL? 👍 👎

◆ Agentic Infrastructure

HeyGen HyperFrames

Feedback ⓘ v0.8.202 NOTES ↗ RANK

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

> HyperFrames shipped a whole-number scale flag for Lambda rendering, a contrast-check exclusion option, faster Lambda deploys via function warming, plus a Studio timeline zoom action and a new CSS lint rule.

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

01 **Whole-number scale rendering for Lambda render** NEW — 90

Adds a `--width` / `--height` flag to `lambda render` that renders at the specified whole-number multiple scale, e.g. `hyperframes lambda render --width 2 --height 2` renders at exactly 2x base resolution without manually specifying pixel values.

Render a Lambda video at exactly 2× its base resolution without specifying pixel values manually.

```
$ hyperframes lambda render --width 2 --height 2
```

— Named flag with runnable example command v0.8.202

THINNER COVERAGE BELOW

02 **Lambda deploy warms video function** IMPROVED — 45

Lambda deploy now warms the video function so the first render does not wait for its image to load.

— Automatic behaviour change, no user action required v0.8.202

03 **Studio timeline zoom to selection** NEW — 40

Adds a Studio timeline action to zoom to the currently selected clips.

— UI action named but no exact navigation path given v0.8.201

04 **Lint check for inline style selectors** NEW — 40

Adds a Lint check that flags CSS rules targeting the inline `style` attribute.

— Names the attribute checked but no invocation shown v0.8.201

05 **Contrast check exclusion during animation** NEW — 35

Adds an opt-in check flag to the CLI that excludes contrast measured mid-animation.

— Flag name not given, only described in prose v0.8.202

WAS THIS USEFUL? 👍 👎

◆ Inference Serving & Optimization

OpenNMT CTranslate2 Feedback ⓘ v4.8.4 NOTES ↗ RANK

CTranslate2 runs Transformer models with a C++ inference engine, quantization, and CPU and GPU optimizations.

> CTranslate2 v4.8.4 adds experimental support for Drama 3 text-to-speech models along with the new ops needed to run them, plus Python 3.15 wheel builds.

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

01 **Drama 3 text-to-speech support with new audio ops** NEW 68

Adds an experimental implementation of Drama 3 offline text-to-speech, including model conversion and waveform decoding on CPU with optional CUDA and ROCm acceleration. This is backed by new CPU and GPU implementations of the `CausalDepthwiseConv1D`, `ConvTranspose1D`, `DequantizeFP8`, `GatedDeltaNet`, `MoeTopK`, and `Snake` operations for audio models.

— Names exact ops and acceleration backends but experimental scope unclear v4.8.4

THINNER COVERAGE BELOW

02 **Python 3.15 wheel builds** NEW 30

CTranslate2 now builds Python 3.15 wheels.

— Bare statement with no further detail v4.8.4

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

GrowGraph OntoCast

Feedback ⓘ v0.8.0 CODE ↗ RANK

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

> OntoCast v0.8.0 overhauls ontology versioning with code-minted identity, head/divergence tracking, a strict version-header contract, and a richer HTTP API, while v0.7.0 added multi-document batch processing to `/process` and `/process_unit`.

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

01 **Multi-document batch processing** NEW 95

`/process` and `/process_unit` now accept multiple files in one request, returning `{"results": [{"filename", "status_code", "outcome"}]}` with HTTP 200 while per-document failures surface as `422` / `500` inside the result. A new `MAX_FILES_PER_REQUEST` config (default 20) caps files per request, returning `400 too_many_files` when exceeded. The new `InputDocument` (`filename` , `data`) carrier, exported from `ontocast` , replaces the flat `AgentState.raw_input` field, and `build_agent_states_from_parsed` produces one `AgentState` per document, superseding `build_agent_state_from_parsed` ; the batch envelope is formalized as `ProcessBatchResponse` .

Submit two PDFs in a single request and inspect per-document outcomes from the batch response.

```
$ curl -X POST http://localhost:8000/process \
  -F file=@report_a.pdf \
  -F file=@report_b.pdf
```

Build agent states for each uploaded document using the new batch helper and `InputDocument` type.

```python
from ontocast import InputDocument
from ontocast.api.process_request import
build_agent_states_from_parsed

documents = [
    InputDocument(filename="doc1.pdf", data=open("doc1.pdf",
"rb").read()),
    InputDocument(filename="doc2.pdf", data=open("doc2.pdf",
"rb").read()),
]
states = build_agent_states_from_parsed(parsed_request)
```

— Config, response shape, types and runnable curl/code examples all given `v0.7.0`

02 **Ontology versioning HTTP API** NEW 92

New endpoints manage ontology version lineage over HTTP: `GET /ontologies` returns `iri`, `ontology_id`, `title`, `description`, `versions`, `heads[]` and `diverged`; `GET /ontologies/{ontology_id}/versions` lists version history newest first; `GET /ontologies/{ontology_id}/versions/{key}` returns one version as Turtle; `DELETE /ontologies/{ontology_id}/versions/{key}` tombstones a version and returns new heads; `GET /ontologies/{ontology_id}/diff` returns added and removed N-Triples; `PUT /ontologies/{iri}` uploads a new version under the current head (returning `409` on several heads); and `POST /ontologies` now returns `version_iri` instead of `hash`, answering `409` when the IRI already exists.

Inspect the full version history of a specific ontology to audit lineage before a merge.

```
$ curl -s 'https://<host>/ontologies/my_domain_model/versions' |
python -m json.tool
```

Tombstone a diverged version and retrieve the resulting new heads to decide what to merge.

```
$ curl -s -X DELETE
'https://<host>/ontologies/my_domain_model/versions/<key>'
```

— Every endpoint, method and response field named with curl examples `v0.8.0`

03 **Strict ontology version-header contract** BREAKING 78

Every stored ontology version is now a named graph `<iri>#<key>` whose ontology IRI must carry `owl:versionIRI`, `owl:priorVersion`, `owl:versionInfo`, `vann:preferredNamespacePrefix` / `vann:preferredNamespaceUri`, `dcterms:title` / `rdfs:label`, `dcterms:description`, `dcterms:created`, `dcterms:modified`, and `prov:wasDerivedFrom`; a removed version keeps its header but is typed `oc:RemovedVersion`. Invalid graphs raise `CatalogContractError`, store refusals raise `TripleStoreWriteError`, and model-written `owl:Ontology` headers are dropped and counted via `ontology/model_header_triples_dropped` and `apply_header_triples_dropped`. `Ontology.version_iri`, `prior_version_iris`, and `modified_at` replace `versioned_iri`, `parent_hashes`, and the old meaning of `created_at`; `dcterms:identifier 'hash:...'` and `urn:hash:` parent references are removed from the header.

— Exhaustive schema and errors named but no reader-facing action `v0.8.0`

04 Ontology version metadata in process responses NEW

76

`/process` and `/process_unit` responses now include `metadata.ontology_versions`, with `role` values `created`, `updated`, and `read`; each `ontology_artifacts[]` entry carries `version` and `version_iri` (null when nothing was written). `AgentState` gains `ontology_versions_written`, recording each stored version with `iri`, `version_iri`, `ontology_id`, `version`, `prior_version_iris`, and `role`.

Inspect which ontology versions were written or read during a document-processing run, using the new `ontology_versions_written` field on `AgentState`.

python

```
for v in agent_state.ontology_versions_written:
    print(v['ontology_id'], v['version_iri'], v['role'])
```

— Fields named precisely but shown via a short code snippet only `v0.8.0`

05 Code-minted ontology identity NEW

73

A new `RESOLVE_HOME_ONTOLOGY` config and LLM phase `ontology_identity` has the model propose an `ontology_id`, title and description for a document: a known id extends an existing ontology, any other is minted as `<CURRENT_DOMAIN>/<ontology_id>`, with `ontology_id` constrained to `^[a-z][a-z0-9_-]{1,31}$`. The proposal is returned via the new `OntologyIdentityProposal` response model, alongside a terms-only `FreshOntologyRenderReport`.

— Mechanism and naming rule given but no worked example `v0.8.0`

06 Ontology head and divergence tracking

BREAKING

68

`OntologyManager.get_head`, `get_head_by_iri`, `get_heads`, and `get_heads_by_iri` replace the previous `get_freshest_terminal_ontology*` and `get_terminal_ontologies*` accessors. An ontology with multiple heads raises `DivergedOntologyError` and is excluded from extraction until merged.

— Named methods and error but no usage example `v0.8.0`

└──▷ BREAKING ON UPGRADE

- ! `AgentState.ontology_updates_by_iri` replaces `ontology_updates_applied`; `ontology_versions_written` is a new field.
- ! `Ontology.version_iri`, `prior_version_iris`, and `modified_at` replace `versioned_iri`, `parent_hashes`, and the old meaning of `created_at` (which now means the root's creation time).
- ! `OntologyManager.get_head`, `get_head_by_iri`, `get_heads`, and `get_heads_by_iri` replace `get_freshest_terminal_ontology*` and `get_terminal_ontologies*`.
- ! `POST /ontologies` returns `version_iri` instead of `hash`.
- ! `dcterms:identifier 'hash:...'` and `urn:hash:` parent references are removed from the version header.
- ! `FreshOntologyRenderReport` replaces `OntologyRenderReport`; `OntologyIdentityProposal` replaces the previous home-ontology response model.
- ! `CURRENT_DOMAIN` must now be an absolute http(s) IRI with no fragment, query string, or trailing slash.
- ! Untyped instances that cannot be repaired against the ontology context are silently dropped (counted as `facts_untyped_dropped`) rather than retained.
- ! `/process` and `/process_unit` now return `{"results": [{"filename", "status_code", "outcome"}]}` (HTTP 200) instead of a single-document response; callers that inspect the top-level status code for document failures will break.
- ! Uploaded files are now identified by their filename rather than their form field name; clients that relied on form-field naming for routing will see changed behaviour.
- ! `AgentState.raw_input` is removed and replaced by `input_document: InputDocument` (`filename`, `data`); any code reading `raw_input` will break.
- ! `build_agent_state_from_parsed` is removed and replaced by `build_agent_states_from_parsed`; call sites using the old function will break.
- ! `SharedEncoder` no longer exposes `.model` or accepts the `serialize=` argument; use `encode`, `token_lengths`, and `max_seq_length` instead.

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ snapshot-20261026 NOTES ↗

RANK

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

> LangSmith let deployments switch GitHub repositories in place, added AI-powered search to tracing project pages, and introduced a metadata filter for thread search, alongside several breaking changes to deployment UI, Teams API access, and deletion behavior.

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

01 Exclude automation threads from thread search NEW

90

`POST /threads/search` now supports an `exclude_metadata` field, letting you filter out automation-generated threads (e.g. from the Open SWE sidebar) by metadata key/value such as `{"source": "automation"}`.

Exclude automation threads (e.g. from Open SWE sidebar) when searching threads in a deployment.

```
$ curl -X POST 'https://api.smith.langchain.com/threads/search' \
  -H 'x-api-key: <your-api-key>' \
  -H 'Content-Type: application/json' \
  -d '{"exclude_metadata": [{"source": "automation"}]}'
```

— Endpoint, field, and runnable example provided. snapshot-20261026

02 AI search on tracing project pages NEW

60

Trace project pages gain an AI search option in the filter bar: select it, enter a plain-language query, and apply it alongside structured filters, date range, and tab selection. Organization admins enable it in organization settings.

— Describes behavior and enablement path but no API or flag. snapshot-20261026

03 Deployment deletion now immediate with confirmation BREAKING

60

Deleting a deployment now removes its database and metadata immediately (previously after a delay) and requires typing permanently delete to confirm.

— Clear before/after behavior and required confirmation phrase. snapshot-20261026

THINNER COVERAGE BELOW

04 **Custom headers relocated to Connect dialog** BREAKING

45

Custom headers for deployments are no longer shown in the Settings dialog — they are now stored and managed only in the Connect dialog.

— Clear UI relocation but no further mechanism described. snapshot-20261026

└──▷ **BREAKING ON UPGRADE**

! The separate Revisions tab on the Deployment overview is removed; bookmarks to tabs after Overview now open the tab to their right instead.

! Teams APIs return `403` unless the organization has the `can_use_teams` setting enabled.

! Custom headers for deployments are no longer shown in the Settings dialog — they are now stored and managed only in the Connect dialog.

! Deleting a deployment now removes its database and metadata immediately (previously after a delay) and requires typing `permanently delete` to confirm.

WAS THIS USEFUL? 👍 👎

UK AI Security Institute Inspect AI Feedback ⓘ

0.3.285 CODE ↗ RANK

Inspect is the UK AI Security Institute's framework for large language model evaluations, covering datasets, solvers, scorers, and agentic eval tasks.

> Inspect AI 0.3.285 reorganizes the eval viewer, adding a Terminals tab for replaying recorded terminal sessions and consolidating prior info views into a single Info tab.

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

01 **Terminals tab and merged Info tab in viewer** NEW

45

Adds a `Terminals` tab in the eval viewer for browsing recorded terminal sessions, and merges the previous info views into a single `Info` tab.

— Names the two new/merged UI tabs but no further mechanism detail. 0.3.285

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ snapshot-20261026 NOTES ↗ RANK

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

> Braintrust shipped GitHub source access inside Loop for trace investigations, extended Loop to dashboards for natural-language chart work, and added Cursor CLI session tracing and structured application logging to the SDK.

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

01 **Structured application logging API** NEW 78

The TypeScript SDK adds `wrapConsole` to forward all existing console output and a `logger` object (`logger.info`, `logger.error`, etc.) to emit structured log entries at different severities, tied to the active trace for side-by-side inspection.

Emit structured application log entries at different severities in TypeScript and tie them to the active trace for side-by-side inspection.

typescript

```typescript
import { wrapConsole, logger } from 'braintrust';

// Forward all existing console output
wrapConsole();

// Or emit directly at a chosen severity
logger.info('Prompt constructed', { tokens: 512 });
logger.error('Scorer timeout', { scorer: 'relevance', elapsed_ms:
5000 });
```

— Named functions with runnable code sample, minor detail on storage snapshot-20261026

THINNER COVERAGE BELOW

02 **GitHub repo integration in Loop** NEW 50

Loop now has a GitHub repo integration (public preview) giving read-only access to source code during trace investigations, with an optional write-access mode that lets Loop open PRs.

— Describes scope and modes but no setup steps or config key snapshot-20261026

03 **Loop on dashboards** NEW 49

Loop can now be used directly on dashboards, enabling natural-language investigation and chart creation or refinement from an individual dashboard or the chart editor.

— Names UI entry points but no commands or config snapshot-20261026

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Open-Weight Models

StandardThinking StandardOne-35B-SH Feedback ⓘ

released · Oct 26, 2026 MODEL ↗

StandardOne-35B-SH from StandardThinking was released on 26 Oct 2026. Open weights on Hugging Face.

WAS THIS USEFUL? 👍 👎

Z.ai GLM 4.5V Feedback ⓘ retirement scheduled · Oct 25, 2026

MODEL ↗

GLM 4.5V is a multimodal model from Z.ai.

OpenRouter lists a retirement date of 31 Dec 2026 for z-ai/glm-4.5v, first seen 25 Oct 2026.

WAS THIS USEFUL? 👍 👎

MiniMax M2.5 Feedback ⓘ price increase · Oct 25, 2026 MODEL ↗

MiniMax M2.5 is a reasoning model from MiniMax.

Price increase on OpenRouter (minimax/minimax-m2.5): $0.27 → $0.30 in, $0.95 → $1.20 out per million tokens (+26%), first seen 25 Oct 2026.

WAS THIS USEFUL? 👍 👎

Inference.net

Feedback ⓘ announced · Oct 23, 2026 BLOG ↗

RANK

Inference.net is a inference infrastructure for AI-native teams. Run, monitor, and optimize production AI workloads with lower cost, faster latency, and dedicated support from Inference.net.

> Inference.net launched MiniMax M3.1 Flash as a new serverless model with a 1M-token context window and selectable reasoning effort.

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

01 **MiniMax M3.1 Flash model launch** NEW 85

Adds `minimax-m3.1-flash` as a new model on the OpenAI-compatible API at `https://api.inference.net/v1`, supporting a 1M-token context window, image input, function calling, structured outputs, and automatic prompt caching. Requests can set a `reasoning_effort` parameter with five selectable levels: `low`, `medium`, `high`, `xhigh`, and `max`.

— Names model id, endpoint, and parameter values; no benchmarks or pricing given.

launch-20261023-a7814531

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · October 26, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue