THE AI TOOLCHAIN

NO. 038

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED SEPTEMBER 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 268 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 40 tools

▾ // THE BRIEF

HIDE SAME ISSUE, SAME PROMPT, TWO WRITERS: Opus 5 | GPT-5.5

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

Three unrelated things stand out today. Anthropic shipped Claude Opus 5.5 for long-running agentic coding, plus context compaction and

vim ? help

cache diagnostics so you can see why a prompt missed cache. Fireworks AI's FireRouter picks a model per turn behind one stable ID. And RunPod finally lets you exercise a serverless handler over local HTTP instead of editing static test inputs.

BUILD

Point your existing coding harness at open models without changing the harness

FireConnect and TogetherLink both aim Claude Code, Cursor and friends at hosted open models, and FireRouter's stable ID picks the best open or closed model per user turn. You get provider choice without rewriting the agent loop or pinning a model ID into your config.

Fireworks AI · Together AI

BUILD

Interrupt an agent mid-call instead of waiting for it to finish being wrong

Codex CLI's opt-in `instant_interrupt` lets new user input preempt an in-flight code-mode call; mlx-serve adds mid-run agent steering and Pipecat tightens interruption handling for voice. The old pattern — watch a long tool call go down a bad path, kill the session, restart from scratch — stops being the only option.

OpenAI Codex CLI · mlx-serve · Pipecat

EVALUATE

Find the jailbreak and the wasted agent steps before your users do

Engine v2 adds proactive Red Teaming for agents in private beta, plus trajectory-inefficiency detection that flags runs taking more steps than the task needs, and performance trends from production traffic. Attack probing and loop detection move from a manual audit into the same place the traces already live.

LangSmith

Does **Opus 5** read better? 👍 👎

OFFENSIVE SECURITY

◆ Exploitation & C2

xalgorix

Feedback ⓘ v4.6.99 NOTES ↗ RANK

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

> xalgorix added a settings UI toggle for controlling auto-delegate behavior without editing environment variables.

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

01 **UI toggle for auto-delegate control** NEW 55

Exposes `XALGORIX_DISABLE_AUTO_DELEGATE` in the web settings UI, allowing auto-delegate behavior to be toggled without editing environment variables directly.

— Names exact config key and UI location, but no further mechanism detail. v4.6.99

v4.6.99 NOTES ↗ LATEST STABLE

Adds `XALGORIX_DISABLE_AUTO_DELEGATE` control to the xalgorix settings UI.

└──▷ GET THIS VERSION

```
$ git clone --branch v4.6.99
https://github.com/xalgorix/xalgorix.git
# already have the repo? check out this version:
$ git checkout v4.6.99
```

WAS THIS USEFUL? 👍 👎

BUILD

◆ AI Coding Agents

Anysphere Cursor

Feedback ⓘ snapshot-20260926 NOTES ↗

RANK

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

> Cursor's biggest additions this window are Security Review, an automated PR vulnerability scanner, and Rollouts, which verifies deployments per-environment — alongside a batch of new Grok Bot settings for network routing, cloud computer management, and per-bot identity.

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

01 Traffic routing through host desktop for Grok Bot NEW

75

Adds a Route traffic through this computer toggle in the Network section of settings, routing the Grok Bot computer's web traffic through the local desktop so destinations see the desktop's IP and the Bot can reach networks only available from that device.

Route a Grok Bot through your desktop's network so it can reach internal resources (e.g. a private SIEM or on-prem scanner) that are only accessible from your machine.

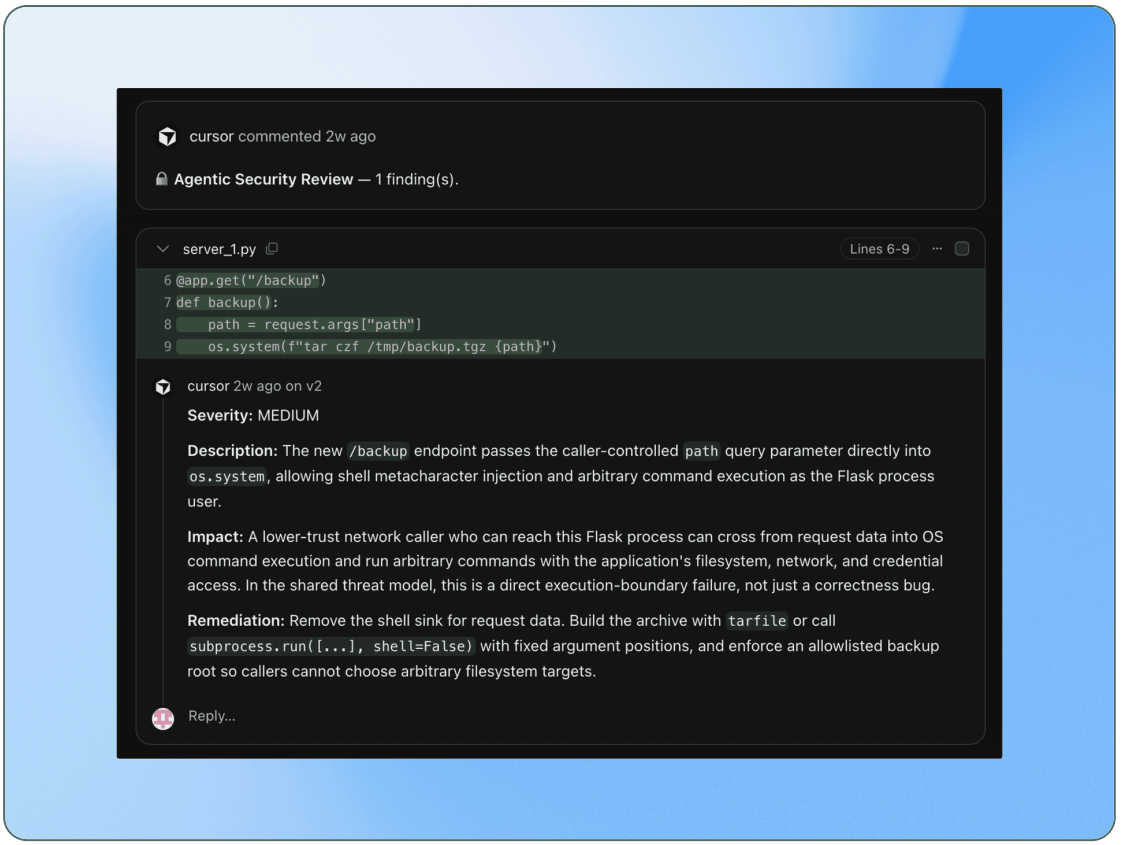
> 📍 In Cursor, open Settings › Network and turn on Route traffic through this computer. The setting applies to new connections immediately — the Bot's outbound traffic will now exit via your desktop's IP.

— Named toggle and effect, exact UI path given product docs

02 Security Review automated PR vulnerability scanning NEW

73

Security Review detects injection across SQL, command, and template surfaces; authentication and authorization bypasses; secrets and credentials committed to source; SSRF and unvalidated redirects; unsafe deserialization; and dependency changes that introduce known vulnerabilities. Findings dismissed with a stated reason are suppressed for that PR and not raised again, and teams can add codebase-specific rules (e.g. requiring all external API calls go through a specific client) that are enforced on every review.



Enable Security Review to automatically scan every non-draft PR for exploitable bugs and enforce team-specific codebase rules on each review.

📍 In the Cursor dashboard, go to the Automations tab › enable Security Review › select the repositories you want reviewed. Optionally add team rules (e.g. 'all external API calls must go through the HttpGateway client') to enforce codebase-specific policies on every PR.

— Names detection categories and dismissal behavior; enable path only via UI

`snapshot-20260926`

---

03 Rollouts per-environment deployment monitoring NEW

70

Rollouts tracks each environment separately so a change can be verified healthy in staging and still flagged in production, checking error and latency signals alongside the change's intended effect. It is enabled from the Automations tab by connecting source control (Origin or GitHub), a deploy system, and a telemetry provider (e.g. Datadog), attaching a monitoring plan to the next pull request opened.

Enable Rollouts for your repository to automatically monitor every PR deployment across staging and production environments.

📍 In the Cursor dashboard, go to the Automations tab › enable Rollouts › connect your source control (Origin or GitHub), deploy system, and telemetry provider (e.g. Datadog). Rollouts will attach a monitoring plan to the next pull request opened.

— Describes mechanism and setup steps but no config keys or API `snapshot-20260926`

---

04 **Grok Bot cloud computer update and reset controls** NEW

65

Adds an `Update Grok Bot's Computer` action in the `Grok Bot's Computer` settings section to rebuild the cloud computer on the latest image while preserving durable state, and a separate `Reset Grok Bot's Computer` action to reset the cloud computer independently of the Grok Bot app.

Rebuild a Grok Bot's cloud computer to a clean latest image after a suspected compromise or configuration drift, without losing persisted state.

> 📍 In Cursor, open Settings › Grok Bot's Computer and click `Update Grok Bot's Computer`. The cloud computer is reprovisioned on the latest image; durable state is preserved.

— Names both actions and their effect; thin beyond that `product docs`

THINNER COVERAGE BELOW

05 **Per-bot identity editing** NEW

51

Clicking the Bot's name at the top of chat and choosing `Bot settings` lets you edit that Bot's name, label, description, notifications preference, and avatar.

— Names editable fields and UI path but no deeper mechanism `product docs`

└──▷ ALSO FROM THESE RELEASES

`snapshot-20260926` NOTES ↗ LATEST STABLE

Cursor: Rollouts tracks each environment separately so a change can be verified healthy in staging and still flagged in production, checking error and latency signals alongside the change's intended effect.

└──▷ HOW TO FIND IT

Enable Rollouts for your repository to automatically monitor every PR deployment across staging and production environments.

> 📍 In the Cursor dashboard, go to the Automations tab › enable Rollouts › connect your source control (Origin or GitHub), deploy system, and telemetry provider (e.g. Datadog). Rollouts will attach a monitoring plan to the next pull request opened.

Enable Security Review to automatically scan every non-draft PR for exploitable bugs and enforce team-specific codebase rules on each review.

📍 In the Cursor dashboard, go to the Automations tab › enable Security Review › select the repositories you want reviewed. Optionally add team rules (e.g. 'all external API calls must go through the HttpGateway client') to enforce codebase-specific policies on every PR.

Security Review detects injection across SQL, command, and template surfaces; authentication and authorization bypasses; secrets and credentials committed to source; SSRF and unvalidated redirects; unsafe deserialization; and dependency changes that introduce known vulnerabilities.

Dismissed Security Review findings with a stated reason are suppressed for that PR and not raised again.

WAS THIS USEFUL? 👍 👎

Red Hat ripwire

Feedback ⓘ v0.6.4 NOTES ↗ RANK

ripwire gives coding agents a ranked, deterministic map of a repository — call graph, blast radius, and tests to run — from a CLI or over MCP.

> ripwire 0.6.4 ships Windows x64 preview builds with matching diagnostics and skill-install guidance, strengthens TypeScript/JavaScript test-runner detection (node:test parsing, version-aware `--experimental-strip-types`, and unconditional run_unknown for .tsx/.jsx), and adds partial-graph disclosure for unresolved TS/JS imports across `--deps`, `--arch` and `--impact`; it also renames `--readability` to `--biggest-first` and adds an untested_modscope= attribute.

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

01 **Windows x64 builds and diagnostics** NEW 95

Adds Windows x64 preview builds as `ripwire-<version>-windows-x64.zip` (with accompanying `.zip.sha256`), built with clang-cl against the static C runtime — no Visual C++ Redistributable required, though an AVX2 CPU is needed. Windows-specific `--doctor` diagnostics report `binary-path` as `degraded="1"` on Windows, surface the first `ripwire` on `Path` via `which=` / `which_version=`, and fall back to `%USERPROFILE%\ripwire-` for cache location when `TMPDIR`, `TEMP`, and `TMP` are all unset. Windows agent skill installation is documented via `bash skills/install.sh` (or `bash skills/install.sh --codex` for Codex) from Git Bash, plus a PowerShell manual copy path targeting `$HOME\.claude\skills` (Claude Code) or `$HOME\.agents\skills` (Codex).

Download and verify the Windows x64 preview binary in PowerShell, then add it to your user PATH for immediate use.

```
$ $v = "0.6.3"; $a = "ripwire-$v-windows-x64"; $u =
"https://github.com/redhat-et/ripwire/releases/download/v$v"
Invoke-WebRequest "$u/$a.zip" -OutFile "$a.zip"; Invoke-
WebRequest "$u/$a.zip.sha256" -OutFile "$a.zip.sha256"
if ((Get-FileHash "$a.zip" -Algorithm SHA256).Hash -eq (Get-
Content "$a.zip.sha256").Split(" ")[0]) { Expand-Archive "$a.zip"
-DestinationPath "$env:LOCALAPPDATA\Programs" } else { throw
"SHA-256 mismatch: do not run $a.zip" }
$bin = "$env:LOCALAPPDATA\Programs\$a"
[Environment]::SetEnvironmentVariable("Path", "$bin;" +
[Environment]::GetEnvironmentVariable("Path", "User"), "User")
$env:Path = "$bin;$env:Path"
```

Download and verify the Windows x64 preview binary in PowerShell before adding it to PATH.

```
$ $v = "0.6.4"; $a = "ripwire-$v-windows-x64"; $u =
"https://github.com/redhat-et/ripwire/releases/download/v$v"
Invoke-WebRequest "$u/$a.zip" -OutFile "$a.zip"; Invoke-
WebRequest "$u/$a.zip.sha256" -OutFile "$a.zip.sha256"
if ((Get-FileHash "$a.zip" -Algorithm SHA256).Hash -eq (Get-
Content "$a.zip.sha256").Split(" ")[0]) { Expand-Archive "$a.zip"
-DestinationPath "$env:LOCALAPPDATA\Programs" } else { throw
"SHA-256 mismatch: do not run $a.zip" }
```

Verify all diagnostic rows are healthy on Windows after installation.

```
$ ripwire . --doctor
```

— Full binary path, build details, and exact PowerShell/doctor commands given.

product docs

02 **Partial-graph disclosure for unresolved TS/JS imports** NEW

80

Adds `imports_unresolved="N"` and `graph_partial="1"` attributes to `--deps` and `--arch` output when TS/JS imports via `paths` aliases, `baseUrl` `-relative` paths, or workspace package names cannot be resolved, preventing silent false acyclic conclusions. Also adds an `imports_unresolved=` field alongside `importers=` in `--impact` output (XML, `--json`, `--format=columnar`, and the MCP `impact` tool) when a TS/JS import could land on one of the symbol's files.

— Names exact attributes, commands and output formats affected. v0.6.4

03 **untested_modscope= root attribute** NEW

60

Adds `untested_modscope=` root attribute to count untestable module-scope owners separately from `untested=`, surfacing why `untested=` may read zero when only untestable entrypoints are in the blast radius.

— Named attribute and rationale given, but no usage example. product docs

THINNER COVERAGE BELOW

04 **Astro frontmatter indexing** NEW

10

ripwire v0.6.4 adds Astro frontmatter indexing, per the release summary; no further mechanism or scope is described.

— Named only in the summary line, with no supporting detail. v0.6.4

v0.6.4 NOTES ↗ LATEST STABLE

ripwire v0.6.4 adds Astro frontmatter indexing, node:test import-based runner derivation, and partial-graph disclosure for unresolved TS/JS aliases

└──▷ GET THIS VERSION

```
$ git clone --branch v0.6.4 https://github.com/redhat-et/ripwire.git
# already have the repo? check out this version:
$ git checkout v0.6.4
```

Adds `imports_unresolved="N"` and `graph_partial="1"` attributes to `--deps` and `--arch` output when TS/JS imports via `paths` aliases, `baseUrl` `-relative` paths, or workspace package names cannot be resolved, preventing silent false `acyclic` conclusions.

Adds `imports_unresolved=` field alongside `importers=` in `--impact` output (XML, `--json`, `--format=columnar`, and MCP `impact` tool) when a TS/JS import could land on one of the symbol's files.

Derives `node --test` runner from a file's own `import test from "node:test"`, `import { test, describe } from "node:test"`, or require("node:test") when `package.json` evidence is absent or inconclusive, using a real parse rather than substring scan.

Emits a Node-version-aware `--experimental-strip-types` flag for `.ts` / `.mts` / `.cts` test commands based on `engines.node` from the nearest manifest, and stays `run_unknown="1"` when the declared range admits any Node below 22.6.

WAS THIS USEFUL? 👍 👎

The Open Engine Zeroshot

Feedback ⓘ v10.9.0 NOTES ↗

RANK

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

> Zeroshot v10.9.0 adds resumable runs with encrypted checkpoint storage and run-scoped environment preparation for shared workspaces.

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

01 Checkpoint resume for failed runs NEW

85

`zeroshot resume RUN_ID` restarts failed local and direct-target runs from a saved node or concurrent group, restoring its input workspace and completed prerequisite outputs. Checkpoints are stored in an encrypted, deduplicated Restic repository per recovery lineage; successful lineages delete their checkpoint repository while failed or lost runs retain it for recovery.

Resume a failed run from its last checkpoint rather than starting the entire graph over — useful after a transient outage or agent timeout.

```
$ zeroshot resume RUN_ID
```

— Names command, storage mechanism, and lineage retention behaviour. v10.9.0

02 Run-scoped environment preparation for shared workspaces NEW

75

New `--environment FILE` and `--no-environment` CLI flags, plus an `environment` field for Docker and hosted targets, attach run-scoped environment inputs to shared workspaces.

Supply a custom environment file when launching a run so that workers and reviewers share a consistently prepared workspace.

```
$ zeroshot run --title 'Add JSON output with tests' --template
software-change --ship --input ./input.json --runtime-config
./runtime.json --environment ./env.json
```

— Names exact flags and target field but not internal mechanism. v10.9.0

v10.9.0 NOTES ↗ LATEST STABLE

Zeroshot v10.9.0 adds checkpoint resume for failed runs and run-scoped environment preparation for shared workspaces.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v10.9.0 https://github.com/the-open-engine/zeroshot.git
# already have the repo? check out this version:
$ git checkout v10.9.0
```

└──▷ **TRY IT**

Resume a failed run from its last checkpoint rather than starting the entire graph over — useful after a transient outage or agent timeout.

```
$ zeroshot resume RUN_ID
```

Supply a custom environment file when launching a run so that workers and reviewers share a consistently prepared workspace.

```
$ zeroshot run --title 'Add JSON output with tests' --template software-change --ship --input ./input.json --runtime-config ./runtime.json --environment ./env.json
```

Adds `zeroshot resume RUN_ID` to restart failed local and direct-target runs from a saved node or concurrent group, restoring its input workspace and completed prerequisite outputs.

Adds `--environment FILE` and `--no-environment` CLI flags, plus an `environment` field for Docker and hosted targets, to attach run-scoped environment inputs to shared workspaces.

Stores run checkpoints in an encrypted, deduplicated Restic repository per recovery lineage; successful lineages delete their checkpoint repository while failed or lost runs retain it for recovery.

WAS THIS USEFUL?  

OpenAI Codex CLI

Feedback ⓘ rust-v0.158.0-alpha.15.1

NOTES ↗ CODE ↗ RANK

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

> This window's headline addition is the opt-in `instant_interrupt` flag that lets new user input preempt in-flight code-mode calls, alongside tighter filesystem sandbox policy for approved commands, item-anchor pagination for `thread/items/list`, a new agent-message-board HTTP client, and a handful of TUI and exec-server refinements.

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

01 Item-anchor pagination for thread/items/list IMPROVED

90

`thread/items/list` pagination now accepts a `cursor` value of `{"type":"item","itemId":"<id>"}` with a non-empty `turnId` as an item anchor, enabling precise forward and reverse pagination within a turn's visible history.

Resume a `thread/items/list` page from a specific known item rather than an opaque cursor — useful when replaying or resuming a long conversation thread from a bookmark.

```json
{"cursor": {"type": "item", "itemId": "item-123", "turnId": "turn-456"}}
```

— Exact cursor shape and API surface with runnable example rust-v0.159.0-alpha.2

02 instant_interrupt feature flag for mid-turn steering NEW

80

The opt-in `instant_interrupt` feature flag (disabled by default) lets code-mode `exec` and `wait` calls yield early when new user input arrives, letting Codex receive steering input without waiting for the current cell to finish; when enabled it also preempts in-progress model response streaming and retry backoff, immediately continuing the turn with queued user input.

— Named flag and mechanism given, but no config path or example shown

rust-v0.159.0-alpha.1

03 Filesystem sandbox policy for approved commands IMPROVED

75

`FileSystemSandboxPolicy::for_approved_command` grants root and root-metadata writes while retaining explicit path and glob denials, and restores standard devices after Linux writable-root binds, tightening filesystem sandbox policy for approved commands on Linux and Windows.

— Named API function and behaviour, no direct usage example `rust-v0.159.0-alpha.2`

---

04 **`tui_right_click_paste` config key** NEW

70

New `tui_right_click_paste` config key enables configurable right-click paste behaviour in the fullscreen TUI.

Enable right-click paste in the fullscreen TUI so you can paste clipboard content without a keyboard shortcut.

```yaml
tui_right_click_paste: true
```

— Config key given with exact YAML example to set it `rust-v0.158.0-alpha.15`

THINNER COVERAGE BELOW

05 **TUI navigation and reasoning shortcut improvements** IMPROVED

55

The `/ps` command now shows multiline command previews; the `Left` key can open the command center from read-only conversations; and reasoning shortcuts can now reach the Max reasoning level.

— Three named UI behaviours grouped, each a clear starting point `rust-v0.158.0-alpha.15`

---

06 **Windows console suppression for code-mode host** IMPROVED

50

Console windows are now suppressed when launching the code-mode host process on Windows by setting `CREATE_NO_WINDOW`.

— Named environment flag but no user-facing action described `rust-v0.159.0-alpha.1`

---

07 **Executor MCP credential boundary preservation** IMPROVED

45

Executor MCP credential boundaries are preserved across reconnects, preventing cached executor MCP declarations from gaining access to host environment credentials.

— Security behaviour described but no surface or trigger to act on `rust-v0.159.0-alpha.1`

08 **Explicit image background control** NEW 15

Codex CLI adds explicit image background control.

— Bare mention in summary with no further detail supplied `rust-v0.158.0-alpha.15.1`

`rust-v0.158.0-alpha.15.1` NOTES ↗ CODE ↗ PRE-RELEASE

Codex CLI adds explicit image background control, opt-in WebSocket auth for exec-server, and a Linux PID namespace flag.

└──▷ GET THIS VERSION

```
$ git clone --branch rust-v0.158.0-alpha.15.1
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.158.0-alpha.15.1
```

└──▷ TRY IT

Run exec-server in a dedicated environment where the sandbox should be able to signal other same-UID processes by inheriting the host PID namespace.

```
$ codex exec-server --linux-sandbox-pid-namespace inherit
```

`rust-v0.159.0-alpha.2` NOTES ↗ CODE ↗ PRE-RELEASE

Codex CLI adds item-anchor pagination to `thread/items/list` and tightens filesystem sandbox policy for approved commands on Linux and Windows.

└──▷ GET THIS VERSION

```
$ git clone --branch rust-v0.159.0-alpha.2
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.159.0-alpha.2
```

└──▷ USE IT

Resume a `thread/items/list` page from a specific known item rather than an opaque cursor — useful when replaying or resuming a long conversation thread from a bookmark.

```json
{"cursor": {"type": "item", "itemId": "item-123", "turnId": "turn-456"}}
```

Extends `thread/items/list` pagination to accept a `cursor` value of `{"type":"item","itemId":"<id>"}` with a non-empty `turnId` as an item anchor, enabling precise forward and reverse pagination within a turn's visible history.

Adds `FileSystemSandboxPolicy::for_approved_command` to grant root and root-metadata writes while retaining explicit path and glob denials, and restores standard devices after Linux writable-root binds.

`rust-v0.159.0-alpha.1` NOTES ↗ CODE ↗ PRE-RELEASE

Codex CLI adds opt-in `instant_interrupt` feature flag so new user input preempts long-running code-mode calls mid-flight.

└──▷ GET THIS VERSION

```
$ git clone --branch rust-v0.159.0-alpha.1
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.159.0-alpha.1
```

Adds the `instant_interrupt` feature flag (disabled by default) to allow code-mode `exec` and `wait` calls to yield early when new user input arrives, letting Codex receive steering input without waiting for the current cell to finish.

Preempts in-progress model response streaming and retry backoff when `instant_interrupt` is enabled, immediately continuing the turn with queued user input.

Suppresses console windows when launching the code-mode host process on Windows by setting `CREATE_NO_WINDOW`.

Preserves executor MCP credential boundaries across reconnects, preventing cached executor MCP declarations from gaining access to host environment credentials.

`rust-v0.158.0-alpha.15` NOTES ↗ CODE ↗ PRE-RELEASE

Codex CLI alpha adds configurable right-click paste in the TUI, multiline command previews in `/ps`, and an HTTP client for remote agent message boards.

└──▷ GET THIS VERSION

```
$ git clone --branch rust-v0.158.0-alpha.15
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.158.0-alpha.15
```

└──▷ **USE IT**

Enable right-click paste in the fullscreen TUI so you can paste clipboard content without a keyboard shortcut.

```yaml
tui_right_click_paste: true
```

Adds `tui_right_click_paste` config key to enable configurable right-click paste behaviour in the fullscreen TUI.

Shows multiline command previews in the `/ps` command output.

Adds `codex-agent-message-board-client` HTTP client implementing the `AgentMessageBoard` interface over a versioned HTTP API, supporting authenticated board creation, deletion, member registration, and board operations with live SSE receivers.

Allows the `Left` key to open the command center from read-only conversations.

Allows reasoning shortcuts to reach the Max reasoning level.

WAS THIS USEFUL? 👍 👎

Cline

Feedback ⓘ desktop-v0.0.37 NOTES ↗ CODE ↗ RANK

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

> Cline desktop v0.0.37 focuses on operational polish, adding an About settings page with update controls and a macOS diagnostics export shortcut.

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

01 **About page with update and highlights controls** NEW

65

A new About page in Settings shows the current version and channel, with Check for updates and Restart to update buttons. It also adds a Highlights shortcut to replay a one-time What's new dialog (shown automatically after an update) that surfaces recent features, plus links to GitHub releases and the full changelog.

— Names UI elements and buttons but no config or API desktop-v0.0.37

THINNER COVERAGE BELOW

02 **macOS diagnostics export menu shortcut** NEW

48

Adds a Help → Export Diagnostics… menu item on macOS that opens the diagnostics export directly, without navigating through Settings.

— Exact menu path given but limited mechanism detail desktop-v0.0.37

desktop-v0.0.37 NOTES ↗ CODE ↗ LATEST STABLE

Cline desktop v0.0.37 adds an About settings page with update controls, a What's new dialog, and a macOS Help menu shortcut for diagnostics export.

└──▷ **GET THIS VERSION**

```
$ git clone --branch desktop-v0.0.37
https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout desktop-v0.0.37
```

Adds a Help → Export Diagnostics… menu item on macOS that opens the diagnostics export directly without navigating through Settings.

Adds a new About page in Settings showing your version and channel, Check for updates and Restart to update buttons, a Highlights shortcut to replay the What's new dialog, and links to GitHub releases and the full changelog.

> Adds a one-time `What's new` dialog shown after an update to surface recent features; accessible again via `Highlights` on the About page.

WAS THIS USEFUL?  

Alibaba Qwen Code

Feedback ⓘ sdk-typescript-v0.1.16

NOTES ↗ RANK

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

> Qwen Code's biggest changes this window are performance work — halved fresh-startup time and a 60% RSS cut — plus new session tooling (`qwen sessions ps`, `/batch-api`) and Managed Runtime support in the Java SDK, alongside several breaking changes including renamed settings and removed tools.

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

01 **Session listing via `qwen sessions ps`** NEW 80

Adds `qwen sessions ps` support for listing managed Agent View sessions alongside regular sessions, letting teams audit all active and managed agent sessions from the CLI.

List all active sessions including managed Agent View sessions to audit running agents from the CLI.

```
$ qwen sessions ps
```

List all active managed Agent View sessions to monitor running agents in a team workflow.

```
$ qwen sessions ps
```

— Runnable command shown with clear use case desktop-v0.24.6 v0.24.6

02 **`/batch-api` slash command** NEW 75

Adds a `/batch-api` slash command that runs an agent-prepared Batch API workflow, letting users kick off batch jobs like triage reports directly against a repository without an interactive session.

Kick off an agent-prepared batch API job against the current repository without entering an interactive session.

```
$ qwen -p "/batch-api process all open issues and produce a triage report"
```

— Runnable command example included `desktop-v0.24.6`

03 **Startup update-check opt-out** NEW 75

Automatic update checks at desktop startup can be disabled by setting `QWEN_DESKTOP_DISABLE_UPDATES=1`, useful in locked-down or air-gapped environments.

Disable the automatic update check at desktop startup — useful in locked-down or air-gapped environments.

```
$ export QWEN_DESKTOP_DISABLE_UPDATES=1
```

— Exact env var and command shown `desktop-v0.24.6`

04 **Per-provider environment variables in settings** NEW 75

Provider-specific environment variables such as `OPENAI_API_KEY` and `OPENAI_BASE_URL` can be set under `settings.env` in the settings file, making them available to every session without exporting them globally in the shell profile.

Set per-provider environment variables in settings so they are available to every session without exporting them globally in your shell profile.

```json
{
  "settings": {
    "env": {
      "OPENAI_API_KEY": "sk-...",
      "OPENAI_BASE_URL": "https://my-proxy.example.com/v1"
    }
  }
}
```

— Concrete config example but no explicit source feature bullet `sdk-typescript-v0.1.16`

05 **`read_many_files` tool removed** BREAKING 75

The `read_many_files` tool has been removed; its functionality moves to a `readManyFiles` utility available only via user `@ -commands`. Any workflow invoking `read_many_files` as a tool will break.

— Exact tool names given with migration path `sdk-typescript-v0.1.16`

06 **Managed Runtime support in Java SDK** NEW 60

The Java SDK gains a managed runtime attestation client that gates Managed Runtime adoption until after attestation, the W0a Managed Workspace binding contract, and a durable Managed Session journal with failover.

— Named contract and client but no usage shown `desktop-v0.24.6`

THINNER COVERAGE BELOW

07 **ACP subagent tracking and model selection** NEW 55

Qwen Code's ACP integration adds `parentToolCallId` and `subagentType` fields for tracking subagent tool calls, and adds model selection through ACP in the VS Code IDE companion.

— Named ACP fields but no usage example given `sdk-typescript-v0.1.16`

08 **Faster startup and lower memory footprint** IMPROVED 45

Fresh-startup time is halved to become typeable, and resident memory (RSS) is cut by 60%.

— Concrete numbers but no benchmark method described `v0.24.6` `desktop-v0.24.6`

09 **Smart Edit tool and ClearcutLogger removed** BREAKING 45

The Smart Edit tool and ClearcutLogger have been removed; integrations relying on the Smart Edit tool will no longer function.

– Named removed components but no replacement given `sdk-typescript-v0.1.16`

10 **Model and reasoning selection at session start in `qwen serve`**

NEW

45

Supports choosing the model and reasoning mode at session creation time when running `qwen serve`.

– Names the command but no example given `desktop-v0.24.6`

11 **Native advisor tool with usage limits** NEW

25

Adds a native advisor tool with consultation behavior and usage limits.

– No detail on limits or invocation `desktop-v0.24.6`

12 **Retry handling for rate-limit and server errors** IMPROVED

20

Retry logic for 429 (rate limit) and 5xx (server) errors has been improved.

– No mechanism or config detail given `sdk-typescript-v0.1.16`

13 **Unified Web UI architecture** NEW

20

Implements a unified Web UI architecture with a shared component library.

– No specifics on components or usage `sdk-typescript-v0.1.16`

14 **Private and group access policies for channels** NEW

20

Adds separate private and group access policies for channels.

– No mechanism or config detail given `desktop-v0.24.6`

15 **Startup benchmark harness script** NEW

10

Adds a startup benchmark harness script.

– Bare mention, no script name or usage `desktop-v0.24.6`

└──▷ **BREAKING ON UPGRADE**

! Negative settings renamed to positive naming: `disable*` settings are now `enable*` — any configuration using the old `disable*` keys will no longer be recognized.

! The `read_many_files` tool is removed; its functionality moves to a `readManyFiles` utility available only via user `@ -commands` — any workflow invoking `read_many_files` as a tool will break.

! The Smart Edit tool and ClearcutLogger are removed — integrations relying on the Smart Edit tool will no longer function.

sdk-typescript-v0.1.16 NOTES ↗ LATEST STABLE

Qwen Code: Adds `parentToolCallId` and `subagentType` fields for ACP subagent tracking.

└──▷ **GET THIS VERSION**

```
$ git clone --branch sdk-typescript-v0.1.16
https://github.com/QwenLM/qwen-code.git
# already have the repo? check out this version:
$ git checkout sdk-typescript-v0.1.16
```

└──▷ **USE IT**

Set per-provider environment variables in settings so they are available to every session without exporting them globally in your shell profile.

```json
{
  "settings": {
    "env": {
      "OPENAI_API_KEY": "sk-...",
      "OPENAI_BASE_URL": "https://my-proxy.example.com/v1"
    }
  }
}
```

Improves retry logic for 429/5xx error handling.

Implements unified Web UI architecture with shared component library.

Adds model selection through ACP in the VS Code IDE companion.

└──▷ **BREAKING ON UPGRADE**

! Negative settings renamed to positive naming: `disable*` settings are now `enable*` — any configuration using the old `disable*` keys will no longer be recognized.

! The `read_many_files` tool is removed; its functionality moves to a `readManyFiles` utility available only via user `@ -commands` — any workflow invoking `read_many_files` as a tool will break.

! The Smart Edit tool and ClearcutLogger are removed — integrations relying on the Smart Edit tool will no longer function.

desktop-v0.24.6 NOTES ↗ STABLE

Qwen Code desktop-v0.24.6 adds screen-share in Live calls, a `/batch-api` workflow, session listing, startup opt-out, and a 60% RSS cut.

└──▷ **GET THIS VERSION**

```
$ git clone --branch desktop-v0.24.6 https://github.com/QwenLM/qwen-code.git
# already have the repo? check out this version:
$ git checkout desktop-v0.24.6
```

└──▷ **TRY IT**

Disable the automatic update check at desktop startup — useful in locked-down or air-gapped environments.

```
$ export QWEN_DESKTOP_DISABLE_UPDATES=1
```

List all active sessions including managed Agent View sessions to audit running agents from the CLI.

```
$ qwen sessions ps
```

Kick off an agent-prepared batch API job against the current repository without entering an interactive session.

```
$ qwen -p "/batch-api process all open issues and produce a triage report"
```

Adds `qwen sessions ps` support for listing managed Agent View sessions alongside regular sessions.

Adds `/batch-api` slash command for an agent-prepared Batch API workflow.

Supports model and reasoning selection at session creation time in `qwen serve`.

Adds separate private and group access policies for channels.

Adds managed runtime attestation client to the Java SDK, gating Managed Runtime adoption until after attestation.

+4 more

v0.24.6 NOTES ↗ STABLE

Qwen Code: Halves fresh-startup time to typeable and cuts RSS by 60%.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.24.6 https://github.com/QwenLM/qwen-code.git
# already have the repo? check out this version:
$ git checkout v0.24.6
```

└──▷ **TRY IT**

List all active managed Agent View sessions to monitor running agents in a team workflow.

```
$ qwen sessions ps
```

WAS THIS USEFUL? 👍 👎

Anthropic Claude Code

Feedback ⓘ v2.1.283 NOTES ↗

CODE ↗ RANK

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

> Claude Code v2.1.283 adds managed controls for restricting which models are available, new observability into gateway requests and tool outputs, a load-testing mode for the Claude apps gateway, and a prompt-audit diagnostic for aging CLAUDE.md files.

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

01 **OTel tool-output logging** NEW 78

Adds `OTEL_LOG_TOOL_CONTENT=1` environment variable to include MCP tool, WebFetch, and WebSearch outputs in the `tool.output` OpenTelemetry span event.

Capture MCP tool and WebSearch outputs in OpenTelemetry traces to give your OTEL backend full visibility into what each tool returned.

```
$ OTEL_LOG_TOOL_CONTENT=1 claude -p 'search for recent CVEs in OpenSSL'
```

— Named env var, span event, and a runnable command v2.1.283

02 **Model access control via availableModelsMatch and deniedModels**

NEW 75

Adds `availableModelsMatch` managed setting, set to `"exact"` to restrict an `availableModels` entry to only the precise model version named, blocking new releases until explicitly listed, plus a `deniedModels` managed setting to block specific models even when `availableModels` would otherwise allow them.

Lock your org to exact model versions so new Claude releases do not silently become available until you explicitly approve them.

```json
{
  "availableModels": ["claude-opus-4-5"],
  "availableModelsMatch": "exact",
  "deniedModels": ["claude-haiku-3"]
}
```

— Two named config keys with a worked JSON example v2.1.283

03 **Prompt-audit diagnostic in /doctor** NEW 75

Adds `/doctor prompt-audit` (also available as `/checkup prompt-audit`) to audit CLAUDE.md files, skills, agents, and commands for prompting patterns written for older models.

Audit your CLAUDE.md files and custom commands for stale prompting patterns before upgrading to a newer model.

```
$ /doctor prompt-audit
```

— Exact runnable command with clear purpose v2.1.283

04 **Gateway prompt hint headers** NEW 64

Adds `CLAUDE_CODE_GATEWAY_HINT_HEADERS=1` environment variable to attach an `x-claude-code-prompt-id` header to gateway hint headers, letting LLM gateways group all requests that serve one user prompt.

— Named env var and header but no worked example v2.1.283

05 **Gateway load-test mode** NEW 61

Adds a `load_test_mode` opt-in block to the Claude apps gateway config: requests are built and signed but not sent upstream, and clients receive a canned reply, enabling load testing without hitting the real API.

— Mechanism is explained but no config example shown v2.1.283

THINNER COVERAGE BELOW

06 **Plugin load-failure path reporting** IMPROVED 45

Adds a `path` field to `--plugin-dir` load-failure entries in the stream-json `system/init` `plugin_errors`, naming the directory that failed to load.

— Names the field and location but no usage shown v2.1.283

v2.1.283 NOTES ↗ CODE ↗ LATEST STABLE

Claude Code v2.1.283 adds prompt-audit diagnostics, gateway load-test mode, model deny-listing, and richer OTel tool-output logging.

└──▷ GET THIS VERSION

```
$ git clone --branch v2.1.283 https://github.com/anthropics/claude-code.git
# already have the repo? check out this version:
$ git checkout v2.1.283
```

└──▷ **TRY IT**

Audit your CLAUDE.md files and custom commands for stale prompting patterns before upgrading to a newer model.

```
$ /doctor prompt-audit
```

Lock your org to exact model versions so new Claude releases do not silently become available until you explicitly approve them.

```json
{
  "availableModels": ["claude-opus-4-5"],
  "availableModelsMatch": "exact",
  "deniedModels": ["claude-haiku-3"]
}
```

Capture MCP tool and WebSearch outputs in OpenTelemetry traces to give your OTEL backend full visibility into what each tool returned.

```
$ OTEL_LOG_TOOL_CONTENT=1 claude -p 'search for recent CVEs in OpenSSL'
```

Adds `CLAUDE_CODE_GATEWAY_HINT_HEADERS=1` environment variable to attach `x-claude-code-prompt-id` to gateway hint headers, letting LLM gateways group all requests that serve one user prompt.

Adds `availableModelsMatch` managed setting: set to `"exact"` to restrict an `availableModels` entry to only the precise model version named, blocking new releases until explicitly listed.

Adds `deniedModels` managed setting to block specific models even when `availableModels` would otherwise allow them.

Adds `OTEL_LOG_TOOL_CONTENT=1` environment variable to include MCP tool, WebFetch, and WebSearch outputs in the `tool.output` OpenTelemetry span event.

> Adds `load_test_mode` opt-in block to the Claude apps gateway config: requests are built and signed but not sent upstream, and clients receive a canned reply, enabling load testing without hitting the real API.
>
> +3 more

WAS THIS USEFUL?  

Letta Code

Feedback ⓘ v0.33.2 NOTES ↗ RANK

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

> Letta Code now automatically syncs agent system prompts with Letta Cloud on resume (with an opt-out), and adds schema validation to enforce structured output from Workflow agents.

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

01 Cloud-managed system prompt sync on resume BREAKING

72

When resuming a session on Letta Cloud, bundled default system prompts (including older versions) are now automatically replaced by Cloud-managed prompts. Set `LETTA_CODE_PRESERVE_CLOUD_SYSTEM_PROMPT=1` to disable this automatic update and preserve a custom or previous prompt.

Prevent Letta Cloud from overwriting your custom system prompt when an agent session resumes — useful when you maintain a hand-tuned prompt that should never be replaced by a bundled default.

```
$ LETTA_CODE_PRESERVE_CLOUD_SYSTEM_PROMPT=1 letta-code
```

Prevent Letta Cloud from overwriting a customised system prompt when resuming an agent session.

```
$ LETTA_CODE_PRESERVE_CLOUD_SYSTEM_PROMPT=1 letta-code
```

– Names exact env var and before/after behaviour, with runnable example. product docs

THINNER COVERAGE BELOW

02 Schema validation for Workflow agent results NEW

30

Workflow agent results are now validated against schemas to enforce structured output contracts.

– States what changed but no schema format or config surface given. v0.33.2

└──▷ **BREAKING ON UPGRADE**

! On resume, bundled default system prompts (including older versions) are now automatically replaced by Cloud-managed prompts; set

`LETTA_CODE_PRESERVE_CLOUD_SYSTEM_PROMPT=1` to retain the previous behaviour.

v0.33.2 NOTES ↗ LATEST STABLE

Letta Code v0.33.2 adds schema validation for Workflow agent results.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.33.2 https://github.com/letta-ai/letta-code.git
# already have the repo? check out this version:
$ git checkout v0.33.2
```

Validates Workflow agent results against schemas to enforce structured output contracts.

WAS THIS USEFUL? 👍 👎

All Hands AI OpenHands

Feedback ⓘ v1.24.0 NOTES ↗

RANK

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

> OpenHands' latest releases focus on OpenHands Cloud's Agent Canvas, adding read-only viewing of shared automation runs and a log viewer for script-only automations, alongside smaller workspace and backend-caching fixes.

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

01 **Read-only shared automation conversations** NEW 73

Organization members on OpenHands Cloud can open a run link from the Activity Log or a `/conversations/<id>` URL and Agent Canvas will render the conversation in read-only mode for non-owners, preventing unintended edits to shared automations.

— Names exact URL pattern and UI surface, but no config/API detail v1.24.0

02 **View logs for script automation runs** NEW 63

Adds a View logs control on script automation runs so practitioners can read a script bundle's output directly in Agent Canvas, since script runs have no conversation to view; usable while a run is still in progress before the sandbox is cleaned up.

Share a running script automation's live output with a teammate before the sandbox is deleted.

📍 In Agent Canvas, navigate to the automation run in the Activity Log, click View logs while the run is in progress to read the script output before the sandbox is cleaned up.

— Clear UI path and use case, but limited mechanism detail product docs

THINNER COVERAGE BELOW

03 **Backend-scoped OpenAI subscription caches** IMPROVED 41

Scopes OpenAI subscription caches by backend, so switching backends no longer serves stale model or subscription data from a different backend.

— Explains the fix but gives no cache key or config surface v1.24.0

04 **Toggle-all control for workspace folders** NEW 40

Adds a toggle-all control in the Conversations header to expand or collapse all workspace folders at once.

— Bare one-line description of a UI control v1.24.0

└──▷ ALSO FROM THESE RELEASES

v1.24.0 NOTES ↗ LATEST STABLE

OpenHands v1.24.0 adds bulk workspace folder toggling, backend-scoped OpenAI caches, and read-only shared automation conversations.

└──▷ GET THIS VERSION

```
$ git clone --branch v1.24.0
https://github.com/OpenHands/OpenHands.git
# already have the repo? check out this version:
$ git checkout v1.24.0
```

Adds a toggle-all control in the Conversations header to expand or collapse all workspace folders at once.

Scopes OpenAI subscription caches by backend, so switching backends no longer serves stale model or subscription data from a different backend.

Opens shared automation conversations in read-only mode on cloud, preventing unintended edits to shared automations.

WAS THIS USEFUL? 👍 👎

Trailhq Graft

Feedback ⓘ v0.20.0 CODE ↗ RANK

Graft builds a code knowledge graph that gives coding agents compact, repository-specific context through a CLI and MCP server.

> Graft v0.20.0 adds a command to pull accepted CLAUDE.md suggestions from Trail, speeds up initial suggestion delivery with an early-upload push stage, and renames the `graft brain` command group to `graft trail`.

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

01 Pull accepted CLAUDE.md changes from Trail NEW

95

The new `graft claude-md pull` command fetches accepted CLAUDE.md changes from Trail via `GET /api/public/brains/:id/claude-md`, applies them line-by-line to the local instruction file, and reports the applied change IDs back to Trail via `POST /api/public/brains/:id/claude-md/applied`.

After reviewing and accepting CLAUDE.md suggestions in the Trail UI, pull them into your local repo's instruction file so your coding agent picks them up immediately.

```
$ graft claude-md pull
```

— Endpoints, command and update flow all named v0.20.0

02 Early-upload stage for faster CLAUDE.md suggestions IMPROVED

80

`graft trail push` now includes an early-upload stage that reads CLAUDE.md/AGENTS.md, the newest 30 closed PR threads, and the newest 80 commits, posting them to `POST /api/public/brains/:id/repo?stage=early` before the full history read completes. This lets Trail return first CLAUDE.md suggestions roughly 9 seconds after push.

— Endpoint and inputs named, no direct usage command v0.20.0

03 Renamed `graft brain` command group to `graft trail` BREAKING

70

The `graft brain` command group is renamed to `graft trail`, covering `graft trail push | pull | status | connect | disconnect`, and `graft init --brain` becomes `graft init --trail`. Old spellings remain functional for backward compatibility via `src/legacy-args.ts`.

Connect a repo to Trail using the new canonical command name instead of the old `graft brain` spelling.

```
$ graft init --trail <id>:<token>
```

— All renamed subcommands listed with compatibility note v0.20.0

v0.20.0 CODE ↗ LATEST STABLE

Graft v0.20.0 adds `graft claude-md pull` to sync accepted CLAUDE.md changes and renames `graft brain` to `graft trail`.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.20.0 https://github.com/trailhq/Graft.git
# already have the repo? check out this version:
$ git checkout v0.20.0
```

└──▷ TRY IT

After reviewing and accepting CLAUDE.md suggestions in the Trail UI, pull them into your local repo's instruction file so your coding agent picks them up immediately.

```
$ graft claude-md pull
```

Connect a repo to Trail using the new canonical command name instead of the old `graft brain` spelling.

```
$ graft init --trail <id>:<token>
```

Adds `graft claude-md pull` command, which fetches accepted CLAUDE.md changes from Trail via `GET /api/public/brains/:id/claude-md` and applies them line-by-line to the local instruction file, then reports applied change IDs back via `POST …/claude-md/applied`.

Adds early-upload stage to `graft trail push`: reads CLAUDE.md/AGENTS.md, the newest 30 closed PR threads, and the newest 80 commits and posts them to `POST /api/public/brains/:id/repo?stage=early` before the full history read completes, enabling Trail to return first CLAUDE.md suggestions ~9 s after push.

Renames the `graft brain` command group to `graft trail` (covering `graft trail push | pull | status | connect | disconnect`) and `graft init --brain` to `graft init --trail`; old spellings remain functional via `src/legacy-args.ts`.

WAS THIS USEFUL?  

Replit Agent

Feedback ⓘ changelog · Sep 26, 2026 CHANGELOG ↗

RANK

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

> Replit Agent's biggest additions this window were in-workflow access requests for admins, inline Atta-powered chart visualizations in chat, and support for Meta device builds and Airwallex payment integrations via MCP.

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

01 **In-workflow access requests for admins** NEW 70

Adds Settings → Advanced → Access requests in Workspace settings so admins can enable in-workflow access requests; admins review, approve, or deny requests via the Admin API using a key with appropriate permissions. Members hitting usage limits or needing role upgrades see a request prompt without leaving Replit.

Enable in-workflow access requests so team members can request usage limit increases or role upgrades without leaving Replit.

> 📍 In Replit, go to Settings → Advanced → Access requests and toggle access requests on for the Workspace. Members hitting limits will see a request prompt; review and act on requests through the Admin API with a key that has the appropriate permissions.

— Names exact UI path and mechanism, but not API endpoint specifics

changelog-20260926-bd6b953a

02 **Inline data-visualization charts in chat** NEW 70

Adds interactive, inline charts directly in Replit chat powered by Atta (acquired by Replit): upload a dataset or connect a data source, ask a question or request a chart type, and Agent returns a visualization without writing SQL. Renders directly inside the conversation without leaving the chat.

When you want to visualize a dataset in chat without writing SQL — just prompt Replit Agent directly.

> 📍 In a Replit chat conversation: upload a dataset or connect a data source, then type: let's visualize this or ask for a specific chart type (e.g. 'show me a waterfall chart of revenue by quarter').

— Explains mechanism and gives a working prompt example launch-20260925-112d81e5

changelog-20260926-bd6b953a

03 **Airwallex payment integration via MCP** NEW 60

Adds Airwallex Developer to the MCP directory so Agent can build and test payment integrations against the Airwallex sandbox without touching production accounts, connected via a Connect via MCP flow alongside an Airwallex setup guide.

Connect Agent to Airwallex via MCP to build and test payment integrations in the sandbox before touching production.

> 📍 In the Replit MCP directory, find Airwallex Developer and follow the Connect via MCP flow together with the Airwallex setup guide. Once connected, prompt Agent to build a payment integration; all test calls run against the Airwallex sandbox account.

— Names the MCP directory entry and sandbox scope with setup steps

changelog-20260926-bd6b953a

THINNER COVERAGE BELOW

04 **New GPT-6 and Claude Opus 5.5 models** NEW 50

Adds GPT-6 Sol, GPT-6 Luna Fast, and `Claude Opus 5.5` as selectable models in Agent settings, with availability gated by tier and Workspace model policies.

— Names specific models but no usage example or steps changelog-20260926-bd6b953a

05 **Meta device build target** NEW 45

Adds a Meta device build target: Agent now generates device-specific instructions and QR-code previews for apps targeting Meta devices, announced at Meta Connect.

— Describes output but no concrete steps or example changelog-20260926-bd6b953a

06 **Workspace permissions for design systems and slide templates** IMPROVED 45

Adds Workspace permission rules to design systems and slide templates — new items are private by default, and published items are accessible to Workspace members.

— States default/published behaviour but no config surface named

changelog-20260926-bd6b953a

07 **Muse integration for spinning up Replit apps** NEW 35

Adds a Replit integration to Muse so conversations in Muse can spin up full Replit apps.

— Brief description, no mechanism or example given changelog-20260926-bd6b953a

changelog · Sep 26, 2026 CHANGELOG ↗ LATEST STABLE

Replit Agent gains Meta device targets, inline charts, Airwallex MCP, new OpenAI/Anthropic models, and in-workflow access requests

└──▷ HOW TO FIND IT

Enable in-workflow access requests so team members can request usage limit increases or role upgrades without leaving Replit.

📍 In Replit, go to Settings → Advanced → Access requests and toggle access requests on for the Workspace. Members hitting limits will see a request prompt; review and act on requests through the Admin API with a key that has the appropriate permissions.

Connect Agent to Airwallex via MCP to build and test payment integrations in the sandbox before touching production.

📍 In the Replit MCP directory, find `Airwallex Developer` and follow the `Connect via MCP` flow together with the Airwallex setup guide. Once connected, prompt Agent to build a payment integration; all test calls run against the Airwallex sandbox account.

Adds `Settings → Advanced → Access requests` in Workspace settings so admins can enable in-workflow access requests; admins review, approve, or deny via the Admin API using a key with appropriate permissions.

Adds Airwallex Developer to the MCP directory so Agent can build and test payment integrations against the Airwallex sandbox without touching production accounts.

Adds GPT-6 Sol, GPT-6 Luna Fast, and Claude Opus 5.5 as selectable models in Agent settings, with availability gated by tier and Workspace model policies.

Adds Meta device build target: Agent now generates device-specific instructions and QR-code previews for apps targeting Meta devices, announced at Meta Connect.

Adds inline charts powered by Atta so data visualizations and reports render directly inside the Replit conversation without leaving the chat.

+2 more

announced · Sep 25, 2026 BLOG ↗

Replit Agent gains interactive, inline data-visualization charts in chat via the Atta acquisition.

└──▷ HOW TO FIND IT

When you want to visualize a dataset in chat without writing SQL — just prompt Replit Agent directly.

📍 In a Replit chat conversation: upload a dataset or connect a data source, then type: `let's visualize this` or ask for a specific chart type (e.g. 'show me a waterfall chart of revenue by quarter').

> Adds interactive charts directly in Replit chat — upload a dataset or connect a data source, ask a question, and Replit Agent returns an Atta-powered visualization without writing SQL.

WAS THIS USEFUL?  

Graphify Labs Graphify

Feedback ⓘ v0.9.69 NOTES ↗

RANK

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

> Graphify expanded its code-to-knowledge-graph extraction to five new languages in this release — OCaml, Elixir, Fortran, Julia, and Kotlin — each with language-specific relation types for methods, protocols, type-bound procedures, enums, and annotations.

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

01 **Julia macro and enum extraction** NEW 54

Extracts Julia macro definitions and `@enum` types — including valued (`red = 1`) and typed (`Color::UInt8`) enum forms — with members connected via the `case_of` relation.

— Names concrete syntax forms and relation but no user action v0.9.69

02 **Fortran derived-type procedure linking** NEW 49

Extracts Fortran derived-type `contains` blocks, linking type-bound procedures to the type and resolving `=> impl` targets.

— Explains resolution mechanism, no usage instructions v0.9.69

03 **Elixir protocol and implementation extraction** NEW 45

Extracts Elixir `defprotocol` / `defimpl` blocks as containers holding their functions, with a same-file `implements` link.

— Describes container and link mechanism, no command shown v0.9.69

04 **Kotlin annotation and property extraction** NEW 45

Extracts Kotlin annotations (class/function/property, use-site targets) and `val` / `var` primary-constructor properties as graph edges.

— Names constructs covered but no runnable example v0.9.69

05 **OCaml method extraction** NEW 41

Graphify extracts OCaml class methods and instance variables as graph edges, using the `method` relation.

— Names the relation but no usage path or limits given v0.9.69

v0.9.69 NOTES ↗ LATEST STABLE

Graphify v0.9.69 expands graph coverage across five languages: OCaml, Elixir, Fortran, Julia, and Kotlin.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.9.69 https://github.com/Graphify-Labs/graphify.git
# already have the repo? check out this version:
$ git checkout v0.9.69
```

Extracts OCaml class methods and instance variables as graph edges, using the `method` relation.

Extracts Elixir `defprotocol` / `defimpl` blocks as containers holding their functions, with a same-file `implements` link.

Extracts Fortran derived-type `contains` blocks, linking type-bound procedures to the type and resolving `=> impl` targets.

Extracts Julia macro definitions and `@enum` types — including valued (`red = 1`) and typed (`Color::UInt8`) enum forms — with members connected via the `case_of` relation.

Extracts Kotlin annotations (class/function/property, use-site targets) and `val` / `var` primary-constructor properties as graph edges.

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

Feynman

Feedback ⓘ v0.5.6 NOTES ↗ RANK

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

> Feynman's biggest work this window is a broad cleanup of legacy tooling — the science workbench, sealed runtime workspace, session-search, and `/feynman-model` command are all removed in favor of stock Pi integration — alongside new ACP editor support, session-lifecycle CLI flags, per-subagent model overrides, and expanded telemetry controls with richer failure diagnostics.

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

01 **Session lifecycle CLI flags** NEW 85

Adds `--resume`, `--session`, `--fork`, `--no-session`, and `--export` flags to control session lifecycle from the CLI, such as resuming a prior session by ID so follow-up prompts share the same context and tool call history.

Resume a previous research session by ID so follow-up questions share the same context and tool call history.

```
$ feynman --session <session-id> --prompt 'Extend the literature review to include 2025 preprints on CRISPR base editing'
```

— Names every flag with a runnable resume example product docs

02 **Per-subagent model override** NEW 85

Supports `subagents.agentOverrides.<name>.model` in `~/.feynman/agent/settings.json` to pin a bundled subagent to a specific model rather than inheriting the main research model.

Pin the bundled `coding` subagent to a specific model rather than inheriting the main research model.

```json
{
  "subagents": {
    "agentOverrides": {
      "coding": {
        "model": "anthropic/claude-opus-5-5"
      }
    }
  }
}
```

— Exact config key, file path, and example given `product docs`

---

03 **Telemetry opt-out and enriched failure data** IMPROVED

85

Telemetry opt-out via `DO_NOT_TRACK=1` (or `FEYNMAN_DO_NOT_TRACK=1`, or `FEYNMAN_TELEMETRY=off`) now covers both CLI and Pi runtime events, checkable with `feynman status`. Telemetry on failures now carries readable error text — the error message and stack trace, the last lines of Pi's error output, a failed tool's error text, and the provider's error message on a failed model call — instead of an opaque hash, with the home folder redacted as `~`.

Check whether telemetry is active, then opt out for an environment where you do not want usage data sent.

```
$ feynman status
export DO_NOT_TRACK=1
```

Opt out of all Feynman telemetry in your shell profile so no usage data is sent to PostHog.

```
$ export DO_NOT_TRACK=1
```

— Names every opt-out variable and the enriched fields `v0.5.6`

---

04 **Browser cookie config keys replace old auth keys** BREAKING

80

The `geminiBrowser`, `allowBrowserAuth`, `browserAuth`, and `chromeProfile` config keys are no longer read; set `allowBrowserCookies` and `browserCookies.profile` in the pi-web-access config (`~/.feynman/agent/settings.json`) instead for Gemini Web cookie access.

— Names exact old and new config keys and file product docs

05 **ACP editor / RPC mode support** NEW 80

Feynman can run as an ACP server for editors like Zed, entering RPC mode without the interactive theme prompt that would otherwise block startup: `PI_ACP_PI_COMMAND=feynman feynman --mode rpc --no-themes`.

Run Feynman as an ACP server for Zed or another ACP editor, bypassing the interactive theme prompt that would otherwise block startup.

```
$ PI_ACP_PI_COMMAND=feynman feynman --mode rpc --no-themes
```

Run Feynman inside an ACP editor like Zed by enabling RPC mode without themes.

```
$ PI_ACP_PI_COMMAND=feynman feynman --mode rpc --no-themes
```

— Gives exact runnable command for editor integration product docs

06 **Provider API key env vars for research search** NEW 75

Reads `EXA_API_KEY`, `PERPLEXITY_API_KEY`, `OPENALEX_API_KEY`, `SEMANTIC_SCHOLAR_API_KEY`, and `NCBI_API_KEY` env vars for web-search and research provider rate limits.

Switch web search to Perplexity and supply its API key so research sessions use your own rate limit.

```
$ feynman search perplexity api-key <your-key>
```

— Names five env vars with a usage example product docs

07 **Guidance for non-interactive setup and login** IMPROVED 70

`feynman setup` and `feynman model login` in non-interactive windows (Git Bash, scripts, AI coding agents) now explain how to proceed via `winpty`, PowerShell, or by setting the provider's API key variable and running `feynman model set`.

— Names exact fallback commands for blocked prompts product docs

08 **Web-search config file relocated** BREAKING 70

Web-search config moves from `~/.feynman/web-search.json` to `~/.feynman/agent/web-search.json`; Feynman migrates the file automatically on first launch, but tooling that writes to the old path will no longer be read.

— Names exact old and new paths and migration step product docs

09 **`/feynman-model` command removed** BREAKING 70

The `/feynman-model` command is removed; use Pi's `/model` for the main model and pi-subagents' `/subagents` command or `subagents.agentOverrides.<name>.model` for per-subagent model selection.

— Names removed command and its two replacements product docs

10 **Broader science database coverage in search tool** IMPROVED 65

`feynman_science_database_search` now covers OpenAlex, arXiv ID lookup, PubMed, Europe PMC (including open-access full-text sections), bioRxiv/medRxiv, and Crossref.

— Names sources covered but no usage example given product docs

11 **Fast-fail on dead network connections** BREAKING 60

Dead network connections now fail in about 15 seconds with 'Connection error.' instead of retrying silently for over five minutes; the old per-launch 6-retry/5-second-backoff connection setting is removed from existing settings on upgrade and replaced with Pi's default fail-fast behavior.

— Explains before/after timing but requires no user action product docs

12 **Sealed runtime workspace removed** BREAKING 60

The sealed runtime workspace (`.feynman/runtime-workspace.tgz`, its hashes, and its lock) is removed and no longer verified on launch; Feynman now uses stock Pi 0.87.1 loaded as an npm dependency.

— Explains mechanism change but no reader action needed product docs

13 **Science workbench and BioNeMo tool removed** BREAKING 60

The `feynman workbench` command and its `--port`, `--no-open`, `--no-auth`, and `--host` flags are removed, along with the tools `feynman_connector_tools`, `feynman_connector_call`, `feynman_workbench_context`, and `feynman_open_chemistry_sketcher`. The `feynman_model_endpoint_call` tool for the NVIDIA BioNeMo endpoint is also removed.

— Names every removed flag and tool, no migration given `product docs`

└──▷ ALSO FROM THESE RELEASES

Raise Ollama's context window and token budget so local models can write output files during long research runs.

```json
{
  "models": [
    {
      "id": "ollama/llama3.3",
      "contextWindow": 32768,
      "maxTokens": 8192
    }
  ]
}
```

Raise Ollama context and cap token output for local models that silently write no files due to a small context window.

```json
{
  "contextWindow": 32768,
  "maxTokens": 4096
}
```

└──▷ BREAKING ON UPGRADE

- ! The `geminiBrowser`, `allowBrowserAuth`, `browserAuth`, and `chromeProfile` config keys are no longer read; set `allowBrowserCookies` (and optionally `browserCookies.profile`) in `~/.feynman/agent/settings.json` instead.
- ! Web-search config moves from `~/.feynman/web-search.json` to `~/.feynman/agent/web-search.json`; Feynman migrates the file on first launch, but tooling that writes to the old path will no longer be read.
- ! The `/feynman-model` command is removed; use Pi's `/model` for the main model and pi-subagents' `/subagents` or `subagents.agentOverrides.<name>.model` for per-subagent model selection.
- ! The `session-search` optional package preset is removed; it never indexed `~/.feynman/sessions` and its `/search` command collided with pi-web-access.
- ! The sealed runtime workspace (`.feynman/runtime-workspace.tgz`, its hashes, and its lock) is removed; Feynman now uses stock Pi 0.87.1 loaded as an npm dependency.
- ! The `feynman workbench` command and its `--port`, `--no-open`, `--no-auth`, and `--host` flags are removed along with the tools `feynman_connector_tools`,

`feynman_connector_call`, `feynman_workbench_context`, and `feynman_open_chemistry_sketcher`.

- ! The old per-launch 6-retry/5-second-backoff connection setting is removed from existing settings on upgrade; connection errors now use Pi's default and fail fast (~15 s).
- ! The `geminiBrowser`, `allowBrowserAuth`, `browserAuth`, and `chromeProfile` config keys are no longer read; set `allowBrowserCookies` and `browserCookies.profile` in the pi-web-access config instead.
- ! `/feynman-model` is removed; use Pi's `/model` for the main model and pi-subagents' `/subagents` or `subagents.agentOverrides.<name>.model` for per-subagent model overrides.
- ! The `session-search` optional package preset is removed; search past sessions with `grep` or `ripgrep` over `~/.feynman/sessions`.
- ! The science workbench (`feynman workbench`) and its `--port`, `--no-open`, `--no-auth`, `--host` flags, and tools `feynman_connector_tools`, `feynman_connector_call`, `feynman_workbench_context`, and `feynman_open_chemistry_sketcher` are removed.
- ! `feynman_model_endpoint_call` (NVIDIA BioNeMo endpoint tool) is removed.
- ! The sealed runtime workspace (`.feynman/runtime-workspace.tgz`, its hashes, and lock) is removed and no longer verified on launch.
- ! The existing network-retry setting (6-retry, 5-second backoff applied to connection errors) is removed from existing settings and replaced with Pi's default.

v0.5.6 NOTES ↗ LATEST STABLE

Feynman v0.5.6 adds readable error text to telemetry and a `FEYNMAN_TELEMETRY=off` opt-out for enriched failure data.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.5.6 https://github.com/Companion-
Inc/feynman.git
# already have the repo? check out this version:
$ git checkout v0.5.6
```

Supports `FEYNMAN_TELEMETRY=off` (or `DO_NOT_TRACK=1`) to opt out of the newly enriched failure telemetry that now includes error messages, stack traces, and provider error text.

Telemetry now carries readable error text — including the error message and stack trace, the last lines of Pi's error output, a failed tool's error text, and the provider's error message on a failed model call — instead of an opaque hash; home folder is redacted as `~`.

WAS THIS USEFUL?  

Daily Pipecat

Feedback ⓘ v1.12.0 NOTES ↗ RANK

Pipecat is a framework for realtime voice and multimodal agents, orchestrating speech-to-text, model inference, and text-to-speech as a streaming pipeline.

> Pipecat's v1.12.0 window adds a new classifiers subsystem (LLM and Jev-backed) powering voicemail detection and UI-aware agents, IPA pronunciation control across major TTS services, out-of-band LLM inference with JSON schema enforcement, cross-session memory via Memcode, a Media over QUIC transport with reconnect, and finer-grained interruption handling.

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

01 **Classifiers subsystem (LLM and Jev)** NEW 96

Adds `BaseClassifier` with typed result types `YesNoQuestion`, `ChoiceQuestion`, `ScoreQuestion`, and `ClassifierError`, plus `LLMClassifier` (answers classifiers using any Pipecat LLM service that implements run_inference(), batching all questions for a state into one call with a `timeout` defaulting to 10s) and `JevClassifier` (TypeSafe's Jev model, exposing ask(), yes_no(), choice(), score() returning typed results `YesNoResult`, `ChoiceResult`, `ScoreResult`; install with `uv add 'pipecat-ai[jev]'` and set `TYPESAFE_API_KEY`; choice questions capped at `JEV_MAX_CHOICE_OPTIONS` (255)). Classifiers emit metrics via an `on_metrics` event: `ProcessingMetricsData` after every call and, for `JevClassifier`, `LLMUsageMetricsData` for token usage.

Ask a lightweight yes/no question mid-call without spending a full LLM conversation turn — useful for routing or guard-rail checks.

python

```
from pipecat.classifiers.llm import LLMClassifier
from pipecat.classifiers.base import YesNoQuestion

classifier = LLMClassifier(llm=your_llm_service)
result = await classifier.ask(YesNoQuestion(question="Is the
caller asking about billing?"))
if result.answer:  # True/False
    ...
```

Classify an ambiguous user pause as complete, short, or long using Jev in a single low-latency request — useful for end-of-turn detection without an LLM round-trip.

```python
import os
from pipecat.classifiers.jev import JevClassifier, ChoiceQuestion

classifier = JevClassifier(api_key=os.getenv("TYPESAFE_API_KEY"))
results = await classifier.choice(
    "I'd like to book a table for, um",
    {
        "turn": ChoiceQuestion(
            instructions="is the user's turn over?",
            options={
                "complete": "the user finished",
                "short": "a brief pause",
                "long": "asked for time",
            },
        )
    },
)
print(results["turn"].choice)  # e.g. 'short'
```

— Names every method, config key, env var and limit v1.12.0

02 **IPA pronunciation transforms for TTS** NEW 95

Adds pronunciation_transform_ipa() classmethod on TTS service classes to map words to IPA pronunciations, replacing matched words with service-native markup (Cartesia `<<phoneme>>` blocks, ElevenLabs SSML `<phoneme>` tags, Deepgram inline pronunciation objects, Inworld/ElevenLabs v3 slash-delimited IPA), supported across Cartesia, ElevenLabs, Inworld, and Deepgram Aura-2. Should be registered last in `text_transforms` so downstream text processors cannot overwrite the inserted phoneme markup; unsupported services silently skip the transform and unrecognized words are reported once at build time and spoken as written.

Correct pronunciation of medical or technical terms in a Cartesia-powered voice pipeline — register the transform last so downstream text processors cannot overwrite the phoneme markup.

```python
pronunciations = {
    "Metformin": "mɛt'fɔɹmɪn",
    "Adalimumab": "ˌædə'lɪmumæb"
}
pronounce =
CartesiaTTSService.pronunciation_transform_ipa(pronunciations)
# pass pronounce as the last transform in your pipeline
tts = CartesiaTTSService(..., text_transforms=[..., pronounce])
```

Ensure a TTS voice pronounces a medical term correctly across any supported service by registering a pronunciation transform last in the pipeline.

```python
from pipecat.services.cartesia import CartesiaTTSService
from pipecat.processors.filters.markdown_filter import
strip_markdown

pronouce =
CartesiaTTSService.pronunciation_transform_ipa({"Metformin": "mɛt
'fɔɹmɪn"})

tts = CartesiaTTSService(
    text_transforms=[
        ("*", strip_markdown),
        ("*", pronounce),  # last, so nothing rewrites its markup
    ],
)
```

— Runnable example plus every service's markup format named `v1.12.0`

03 Out-of-band LLM inference with schema enforcement NEW

95

Adds run_inference() method on LLM processors to execute one-shot LLM inferences outside the pipeline, accepting an `LLMContext` from `pipecat.processors.aggregators.llm_context` and optional `max_tokens`. A `response_schema` argument enforces a JSON schema on replies for OpenAI, Anthropic, and Google (surfaced via `supports_response_schema`); unsupported providers (AWS Bedrock, DeepSeek, older OpenAI models) ignore it with a warning.

Run a quick one-shot LLM call outside the main pipeline — useful for tool pre-processing or side-channel enrichment without wiring a full frame flow.

```python
from pipecat.processors.aggregators.llm_context import LLMContext

context = LLMContext()
context.add_user_message('What is 2+2?')
response = await llm.run_inference(context, max_tokens=64)
print(response)  # '2 + 2 equals 4.'
```

Enforce a JSON response shape from run_inference() so downstream code can parse the result without fragile text extraction.

```python
schema = {
    'type': 'object',
    'properties': {'answer': {'type': 'string'}},
    'required': ['answer'],
    'additionalProperties': False
}
response = await llm.run_inference(context,
response_schema=schema)
# Returns JSON text matching the schema on OpenAI, Anthropic,
Google
```

— Method, parameters and provider support fully named v1.12.0

04 **No-transcript interruption handling** NEW 86

Adds an `empty_user_turn` parameter on `LLMUserAggregatorParams` to control bot behavior when a user interrupts but STT produces no transcript (cough, background noise, unrecognized speech); configurable via `EmptyUserTurnConfig` with `interrupted_prompt`, `idle_prompt`, and `max_consecutive_recoveries` (default 1), or set to `None` to leave such turns unanswered.

Silence the bot after a no-transcript interruption (e.g. a cough) instead of prompting the user to repeat themselves.

```python
params = LLMUserAggregatorParams(empty_user_turn=None)
```

Recover gracefully when a user makes noise that the STT cannot transcribe, prompting them to repeat only when the bot was mid-response.

```python
from pipecat.processors.aggregators.llm_response import
LLMUserAggregatorParams, EmptyUserTurnConfig

params = LLMUserAggregatorParams(
    empty_user_turn=EmptyUserTurnConfig(
        interrupted_prompt="The user may have said something
unclear. Ask them to repeat.",
        idle_prompt=None,  # stay silent if bot was already done
        max_consecutive_recoveries=2,
    ),
)
```

— Config object and defaults with runnable example v1.12.0

05 **Media over QUIC transport with reconnect** IMPROVED 85

Adds MOQTransport and MOQParams for real-time audio over Media over QUIC (MoQ), supporting serve mode and relay client mode with TLS. Client-mode reconnect redials with backoff within `MOQParams.connection_timeout`, detects silent relay loss via traffic counters, emits `on_disconnected` / `on_connected` per session, and fires `on_client_disconnected` only when the client is truly gone; `MOQRunnerArguments.relay_url` dials a relay by full URL (including query string) and takes precedence over `host` / `port` when both are given.

— Names every config field and event but no runnable snippet v1.12.0

06 **UIWorker screen awareness for voice agents** NEW 78

Adds UIWorker to let a voice agent read and act on a client GUI, exposing `screen_tools`, `classifiers`, and job groups over RTVI, with classifier integration methods `which_element`, `check_screen`, `select_elements`, `should_respond`, and `act`; `UIWorker.snapshot` and `UIWorker.selection` expose the latest page snapshot and selected text to subclasses.

— Every method named but no code example given v1.12.0

07 **Scenario config files and eval judge controls** NEW 76

Adds scenario configuration files supporting `scenarios:` lists, file defaults, `user:` and `judge:` blocks, `factory:`, and `!include` directives for both scripted and simulated scenarios. Adds `JudgeVerdict.confidence` (0–1) and classifier support to eval judges via a `factory:` key in the scenario's `judge.eval:` block (accepting either a `BaseClassifier` or an LLM service), plus `allow_continue` on `judge.eval:` / `EvalJudge` to restrict the judge to yes/no answers when set to `false`.

— Config keys named but no worked config example v1.12.0

08 **Voicemail detection for outbound calls** NEW 60

Adds `VoicemailDetector` and `TTSGate` to classify outbound calls as live conversation or voicemail using a classifier, and leave a message when voicemail answers.

Classify an outbound call as live or voicemail at answer time, so your bot can leave a recorded message instead of running the full conversation flow.

python

```
from pipecat.extensions.voicemail import VoicemailDetector,
TTSGate
from pipecat.classifiers.jev import JevClassifier, JevClient

classifier = JevClassifier(client=JevClient(api_key="<key>"))
detector = VoicemailDetector(classifier=classifier)
gate = TTSGate(voicemail_detector=detector)
```

— Names the classes with a runnable setup, thin mechanism product docs

09 **Per-frame interruptibility** NEW 60

Adds an `interruptible` field to every frame (True by default; False for `UninterruptibleFrame` subclasses) so individual frames can be marked to survive or be dropped during interruption without defining a new frame class.

— Clear mechanism, no code example provided v1.12.0

THINNER COVERAGE BELOW

10 **Cross-session memory via Memcode** NEW 53

Adds a Memcode memory service with separate recall and finalized-turn capture processors, providing OAuth-backed cross-session memory for voice agents.

Add cross-session recall to a voice agent so it remembers facts from previous calls via Memcode.

python

```
from pipecat.services.memory.memcode import MemcodeMemory

memory = MemcodeMemory(oauth_token="<token>")
pipeline = Pipeline([transport.input(), memory.recall(), llm,
memory.capture(), transport.output()])
```

— Brief prose description, no config surface named product docs

11 **Read-only AIService.settings inspection** NEW 51

Adds `AIService.settings` read-only property so code outside a service can inspect current settings (e.g., model) without sending a `ServiceUpdateSettingsFrame`.

— Named property and frame it replaces, no example v1.12.0

12 **Language-aware TTS sentence aggregation** IMPROVED 48

TTS sentence aggregation now follows `Settings.language`, defaulting to English when unspecified, using self-contained sentencex rules with no NLTK downloads or tokenizer warm-up.

— Behavior change described, no usage example v1.12.0

13 **New RTVI function-call report level** NEW 48

Adds `RTVIFunctionCallReportLevel.ARGUMENTS`, a reporting level between `NAME` and `FULL` that emits the function name and arguments but no result.

— Named enum value, minimal behavioural detail v1.12.0

v1.12.0 NOTES ↗ LATEST STABLE

Pipecat v1.12.0 adds IPA pronunciation transforms, Jev/LLM classifiers, MOQ reconnect, and per-frame interruptibility

└──▷ **GET THIS VERSION**

```
$ git clone --branch v1.12.0 https://github.com/pipecat-
ai/pipecat.git
# already have the repo? check out this version:
$ git checkout v1.12.0
```

└──▷ **USE IT**

Ensure a TTS voice pronounces a medical term correctly across any supported service by registering a pronunciation transform last in the pipeline.

```python
from pipecat.services.cartesia import CartesiaTTSService
from pipecat.processors.filters.markdown_filter import
strip_markdown

pronouce =
CartesiaTTSService.pronunciation_transform_ipa({"Metformin": "mɛt
'fɔɹmɪn"})

tts = CartesiaTTSService(
    text_transforms=[
        ("*", strip_markdown),
        ("*", pronounce),  # last, so nothing rewrites its markup
    ],
)
```

Classify an ambiguous user pause as complete, short, or long using Jev in a single low-latency request — useful for end-of-turn detection without an LLM round-trip.

```python
import os
from pipecat.classifiers.jev import JevClassifier, ChoiceQuestion

classifier = JevClassifier(api_key=os.getenv("TYPESAFE_API_KEY"))
results = await classifier.choice(
    "I'd like to book a table for, um",
    {
        "turn": ChoiceQuestion(
            instructions="is the user's turn over?",
            options={
                "complete": "the user finished",
                "short": "a brief pause",
                "long": "asked for time",
            },
        )
    },
)
print(results["turn"].choice)  # e.g. 'short'
```

Recover gracefully when a user makes noise that the STT cannot transcribe, prompting them to repeat only when the bot was mid-response.

```python
from pipecat.processors.aggregators.llm_response import
LLMUserAggregatorParams, EmptyUserTurnConfig

params = LLMUserAggregatorParams(
    empty_user_turn=EmptyUserTurnConfig(
        interrupted_prompt="The user may have said something
unclear. Ask them to repeat.",
        idle_prompt=None,  # stay silent if bot was already done
        max_consecutive_recoveries=2,
    ),
)
```

Adds TTSService.pronunciation_transform_ipa() to build a word-to-IPA mapping that any supporting TTS service (Cartesia, ElevenLabs, Inworld, Deepgram Aura-2) applies as service-native markup; register last in `text_transforms` so nothing rewrites the inserted phoneme blocks.

Adds `LLMUserAggregatorParams.empty_user_turn` (`EmptyUserTurnConfig`) to handle turns that end with no transcript — configurable `interrupted_prompt`, `idle_prompt`, and `max_consecutive_recoveries` (default 1) control how the bot recovers.

Adds `LLMClassifier`, which answers `YesNoQuestion`, `ChoiceQuestion`, and `ScoreQuestion` classifiers using any Pipecat LLM service that supports run_inference(), batching all questions for a state into one call with a `timeout` (default 10 s) and raising `ClassifierError` on failure.

Adds `JevClassifier` and the Pipecat Classifiers subsystem: ask(), yes_no(), choice(), and score() return typed results (`YesNoResult`, `ChoiceResult`, `ScoreResult`) via TypeSafe's Jev model; install with `uv add 'pipecat-ai[jev]'` and set `TYPESAFE_API_KEY`; choice questions are capped at `JEV_MAX_CHOICE_OPTIONS` (255) options.

Adds `response_schema` argument to LLMService.run_inference() so OpenAI, Anthropic, and Google enforce a JSON schema on replies; `supports_response_schema` indicates whether a given service can enforce one.

+10 more

WAS THIS USEFUL?  

CopilotKit

Feedback ⓘ v1.74.0 NOTES ↗ RANK

CopilotKit connects AI agents to application interfaces with shared state, tools, and generative UI components.

> CopilotKit v1.74.0 enables agents to draw on learned skills registered across multiple containers and adds engagement analytics for the copilot launcher HUD in the web-inspector.

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

01 **Launcher HUD engagement analytics** NEW 35

The web-inspector now tracks launcher HUD impressions and actions, surfacing engagement analytics for the copilot launcher.

— Names the surface (web-inspector) but no metrics detail v1.74.0

02 **Cross-container learned skill consumption** NEW 30

Agents can now consume learned skills from multiple containers across SDKs, enabling a single agent to draw on skills registered in different CopilotKit containers.

— Describes capability but no config or API surface named v1.74.0

v1.74.0 NOTES ↗ LATEST STABLE

CopilotKit v1.74.0 adds targeted What's New notifications and cross-container learned skill consumption.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v1.74.0
https://github.com/CopilotKit/CopilotKit.git
# already have the repo? check out this version:
$ git checkout v1.74.0
```

Supports consuming learned skills from multiple containers across SDKs, enabling a single agent to draw on skills registered in different CopilotKit containers.

Tracks launcher HUD impressions and actions in the web-inspector, surfacing engagement analytics for the copilot launcher.

WAS THIS USEFUL? 👍 👎

PydanticAI

Feedback ⓘ v2.51.0 NOTES ↗ RANK

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

> PydanticAI v2.51.0 introduces real-time voice/audio support via a new OpenAI GPT-Live model integration and adds visibility into realtime session context usage.

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

01 **Realtime session context window usage** NEW 40

Exposes `context_window_used` on realtime sessions, sourced from GPT-Live's reported ratio or response usage.

— Names the field and its source but no usage example. v2.51.0

v2.51.0 NOTES ↗ LATEST STABLE

PydanticAI v2.51.0 adds OpenAI GPT-Live support via `OpenAILiveModel` and exposes `context_window_used` on realtime sessions.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v2.51.0 https://github.com/pydantic/pydantic-ai.git
# already have the repo? check out this version:
$ git checkout v2.51.0
```

Adds `OpenAILiveModel` to enable real-time voice/audio sessions with OpenAI's GPT-Live API.

Exposes `context_window_used` on realtime sessions, sourced from GPT-Live's reported ratio or response usage.

WAS THIS USEFUL? 👍 👎

AWS Strands Agents

Feedback ⓘ harness-cli/v0.1.4 NOTES ↗

RANK

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

> Strands Agents added Agent.shutdown() for deterministic resource cleanup and a new a2a_client tool for agent-to-agent delegation, expanded its CLI with new subcommands, and enhanced bidirectional streaming with grouped messages and session snapshots — alongside breaking changes to bidi model resolution and the MCP TypeScript client.

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

01 **Agent.shutdown() for deterministic cleanup** NEW 80

Agent.shutdown() provides scope-based cleanup of agent resources, ensuring connections, sessions, and background tasks are released deterministically after a run.

Ensure all agent resources (connections, sessions, background tasks) are released deterministically after a run.

python

```python
from strands import Agent
from strands_tools import calculator

agent = Agent(tools=[calculator])
try:
    agent('What is 42 * 1764?')
finally:
    await agent.shutdown()
```

— Runnable example demonstrates the exact usage pattern. python/v1.57.1

02 **a2a_client vended tool for agent delegation** NEW 70

A new built-in `a2a_client` tool lets one agent delegate a subtask to another agent over the A2A protocol.

Use the built-in `a2a_client` tool to let one agent delegate a subtask to another agent over the A2A protocol.

```python
from strands import Agent
from strands_tools import a2a_client

orchestrator = Agent(tools=[a2a_client])
orchestrator('Use the a2a_client to ask the code-review agent to
review this diff: ...')
```

— Runnable example shows exact delegation call. python/v1.57.1

03 **CLI command additions and refinements** NEW 67

The CLI gains a `strands update` subcommand alongside polished `compact`, `skills`, `rename`, and `setup` flows, and a revamped CLI adds a `/tools` command.

— Names exact CLI subcommands but shows no usage examples. harness-cli/v0.1.4 python/v1.57.1

THINNER COVERAGE BELOW

04 **MCP TypeScript client upgraded to 2.0** BREAKING 55

MCP TypeScript integration swaps to `@modelcontextprotocol/client` 2.0, which may break existing MCP client configurations built against 1.x.

— Names the exact package/version but no migration steps. python/v1.57.1

05 **Explicit model IDs required for bidi streaming** BREAKING 50

Bidirectional streaming now requires explicit model IDs for providers; implicit/default model ID resolution is no longer supported.

— No migration example or affected config key given. python/v1.57.1

06 **Bidirectional streaming enhancements** NEW 40

Bidirectional (bidi) streaming now supports grouped content messages, and gains snapshot capture and restore for bidi sessions.

— No usage example or config detail shown for bidi features. harness-cli/v0.1.4 python/v1.57.1

└──▷ **BREAKING ON UPGRADE**

! Bidirectional streaming now requires explicit model IDs for providers; implicit/default model ID resolution is no longer supported.

! MCP TypeScript integration swaps to `@modelcontextprotocol/client` 2.0, which may break existing MCP client configurations built against 1.x.

harness-cli/v0.1.4 NOTES ↗ LATEST STABLE

Strands CLI gains a `strands update` command; bidirectional streaming adds grouped content messages and snapshot capture/restore.

└──▷ **GET THIS VERSION**

```
$ git clone --branch harness-cli/v0.1.4 https://github.com/strands-agents/harness-sdk.git
# already have the repo? check out this version:
$ git checkout harness-cli/v0.1.4
```

Adds `strands update` subcommand to the CLI, alongside polished `compact`, `skills`, `rename`, and `setup` flows.

Supports grouped content messages in the bidirectional (bidi) streaming layer.

Adds snapshot capture and restore to the bidirectional (bidi) streaming layer.

python/v1.57.1 NOTES ↗ STABLE

Strands Agents v1.57.1 adds snapshot/restore for bidi sessions, `a2a_client` vended tool, Agent.shutdown(), and a revamped CLI with `/tools` command.

└──▷ **GET THIS VERSION**

```
$ git clone --branch python/v1.57.1 https://github.com/strands-agents/harness-sdk.git
# already have the repo? check out this version:
$ git checkout python/v1.57.1
```

└──▷ **USE IT**

Ensure all agent resources (connections, sessions, background tasks) are released deterministically after a run.

python

```python
from strands import Agent
from strands_tools import calculator

agent = Agent(tools=[calculator])
try:
    agent('What is 42 * 1764?')
finally:
    await agent.shutdown()
```

Use the built-in `a2a_client` tool to let one agent delegate a subtask to another agent over the A2A protocol.

```python
from strands import Agent
from strands_tools import a2a_client

orchestrator = Agent(tools=[a2a_client])
orchestrator('Use the a2a_client to ask the code-review agent to
review this diff: ...')
```

Adds Agent.shutdown() method for scope-based cleanup of agent resources.

└──▷ **BREAKING ON UPGRADE**

- ! Bidirectional streaming now requires explicit model IDs for providers; implicit/default model ID resolution is no longer supported.
- ! MCP TypeScript integration swaps to `@modelcontextprotocol/client` 2.0, which may break existing MCP client configurations built against 1.x.

WAS THIS USEFUL?  

Google ADK

Feedback ⓘ v2.10.0 NOTES ↗ RANK

The Agent Development Kit is Google's code-first Python toolkit for building, evaluating, and deploying agents, with built-in tools, sessions, and deployment to Vertex AI.

> Google's Agent Development Kit added experimental skill lifecycle management, expanded OpenAI reasoning-model support, a new MongoDB toolset, and a Model Armor integration to screen agent input/output, alongside breaking changes to BigQuery protected-mode behavior, eval validation, and instruction templating.

└──▷ **WHAT SHIPPED · 20 FEATURES** most completely described first ? what's the number?

01 **OpenAI reasoning model support in OpenAILlm** NEW 95

Adds `OpenAIGenerateContentConfig` with a per-model reasoning `effort` field for OpenAI reasoning models, `api_key` and `base_url` with per-request key resolution on `OpenAILlm`, and a `google-adk[openai]` installation extra. `OpenAILlm` now surfaces streaming usage and finish reason, reports reasoning tokens on Chat Completions, and honors `tool_config` mode and all tool declarations. `OpenAIResponsesLlm` now ignores `thinking_config` (`thinking_level` / `thinking_budget`) with a warning; use `OpenAIGenerateContentConfig.effort` instead.

Use OpenAI reasoning models with precise effort control instead of the deprecated thinking_config.

python

```python
from google.adk.integrations.openai import OpenAILlm,
OpenAIGenerateContentConfig

llm = OpenAILlm(
    model='o3-mini',

generate_content_config=OpenAIGenerateContentConfig(effort='high'
),
)
```

Install the new OpenAI integration extra and use per-request api_key resolution in OpenAILlm.

```
$ pip install "google-adk[openai]"
```

— Runnable pip install and code sample with named config classes v2.10.0

02 **Skill lifecycle and discovery controls in SkillToolset** NEW 90

Adds `ADK_ENABLE_SKILL_LIFECYCLE=1` to enable experimental skill lifecycle management, including an EPHEMERAL lifecycle (tools last one turn), an active-skill cap on `SkillToolset`, an opt-in `unload_skill` tool, and automatic pruning of unloaded skill instructions from later requests. Also adds `SkillDiscoveryMode` to control how the skill catalog is disclosed to the model, programmatic skill activation APIs on `SkillToolset`, and an `on_error` handler for skill listing functions.

— Env var and multiple named APIs give a clear, runnable starting point v2.10.0

03 **BigQueryToolConfig defaults and protected-mode behavior change**

85

BREAKING

Adds `default_project_id` and `default_dataset_id` fields to `BigQueryToolConfig`. Protected write mode now rejects multi-statement scripts, `CALL`, `EXPORT DATA`, and any other non-SELECT statement that the dry run does not place in the session's anonymous dataset, and protected mode now keeps its session in process memory instead of session state, so temporary tables do not survive a restart or carry over to another replica.

— Names exact config fields and describes concrete before/after behavior v2.10.0

04 **Deprecated legacy live-audio modules** DEPRECATED

70

Legacy live-audio modules under `google.adk.flows.llm_flows` (`audio_cache_manager`, `audio_transcriber`, `transcription_manager`) now emit a `DeprecationWarning` on import; test suites running with `-W error` will fail on these imports. Move to `google.adk.live`.

— Names exact modules and gives a clear migration target v2.10.0

05 **ADK eval efficiency metrics and stricter validation** IMPROVED

60

Adds duration, token consumption, and model call count efficiency metrics to ADK eval. `AgentEvaluator.evaluate` and `evaluate_eval_set` now raise `ValueError` when no eval cases are evaluated, instead of passing silently.

— Named methods and metrics but no example output v2.10.0

THINNER COVERAGE BELOW

06 **MongoDB toolset with vector and hybrid search** NEW

55

Adds a MongoDB toolset with vector search and hybrid search capabilities.

— Names the capability but no setup or API detail v2.10.0

07 **Per-invocation service tier control** NEW

50

Adds `RunConfig.service_tier` to expose and control the serving tier per invocation.

— Names the config field but no mechanism or values given v2.10.0

08 **Non-blocking function call behavior for Live tools** NEW

50

Adds a `behavior` field to tool definitions to control Live non-blocking function call behavior.

– Names the field but not its possible values or effects v2.10.0

09 **Custom A2A request context in RemoteA2aAgent** NEW 50

Adds a `context_builder` parameter to `RemoteA2aAgent` for customizing A2A request context.

– Names the parameter with no example of use v2.10.0

10 **Dev-only deploy endpoints for Agent Runtime, Cloud Run, and GKE**

NEW 50

Adds dev-only deploy endpoints for Agent Runtime, Cloud Run, and GKE.

– Names three targets but no endpoint paths given v2.10.0

11 **Session options exposure in api_server REST endpoint** NEW 50

Adds session options exposure in the `api_server` REST endpoint.

– Names the endpoint surface but not the options themselves v2.10.0

12 **ADK identification in MCP handshake and requests** NEW 50

Adds ADK identification in the MCP `clientInfo` handshake and in the User-Agent header of MCP HTTP requests.

– Two named surfaces but no format or version string shown v2.10.0

13 **Instruction templating literal variable syntax** BREAKING 50

Instruction templating now leaves `${var}` and backslash-escaped `\{var}` as written instead of filling them from state.

– Exact syntax named but no migration steps given v2.10.0

14 **Abort signal primitives in invocation context** NEW 45

Adds abort signal primitives to `InvocationContext` and Context.

– Names classes but not the signal API itself v2.10.0

15 **Name-based request processor helpers in BaseLlmFlow** NEW

45

Adds name-based request processor helpers to `BaseLlmFlow`.

— Names the class but not the helper functions v2.10.0

16 **gemini-3.8-flash model option in adk create** NEW

45

Adds `gemini-3.8-flash` to the `adk create` model options.

— Exact model name and command are given v2.10.0

17 **DEFAULT_ROUTE fan-out to multiple workflow nodes** IMPROVED

45

Allows `DEFAULT_ROUTE` to fan out to several workflow nodes.

— Names the routing constant but not the workflow API around it v2.10.0

18 **Model Armor integration for agent I/O screening** NEW

35

ADK adds a Model Armor integration to screen agent input and output against Google Cloud security templates.

— Names the integration but gives no config surface or usage detail product docs

19 **Streaming partial function-call arguments across model adapters**

IMPROVED

35

Streams partial function-call arguments across model adapters.

— No named adapters or mechanism given v2.10.0

20 **Per-feature experimental telemetry enabling** NEW

25

Adds per-feature enabling of experimental telemetry.

— No named flag or feature list given v2.10.0

└──▷ **BREAKING ON UPGRADE**

! `BigQueryToolConfig` protected write mode now rejects multi-statement scripts, `CALL`, `EXPORT DATA`, and any other non-SELECT statement that the dry run does not place in the session's anonymous dataset.

! BigQuery protected mode now keeps its session in process memory instead of session state — temporary tables do not survive a restart or carry over to another replica.

! `OpenAIResponsesLlm` now ignores `thinking_config` (`thinking_level` / `thinking_budget`) with a warning; use `OpenAIGenerateContentConfig.effort` instead.

! `AgentEvaluator.evaluate` and `evaluate_eval_set` now raise `ValueError` when no eval cases are evaluated, instead of passing silently.

! Instruction templating now leaves `${var}` and backslash-escaped `\{var}` as written instead of filling them from state.

! Legacy live-audio modules under `google.adk.flows.llm_flows` (`audio_cache_manager`, `audio_transcriber`, `transcription_manager`) now emit a `DeprecationWarning` on import; test suites running with `-W error` will fail on these imports. Move to `google.adk.live`.

v2.10.0 NOTES ↗ LATEST STABLE

ADK v2.10 adds MongoDB vector search, skill lifecycle management, OpenAI reasoning model support, and richer eval efficiency metrics.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v2.10.0 https://github.com/google/adk-
python.git
# already have the repo? check out this version:
$ git checkout v2.10.0
```

└──▷ **USE IT**

Use OpenAI reasoning models with precise effort control instead of the deprecated thinking_config.

```python
from google.adk.integrations.openai import OpenAILlm,
OpenAIGenerateContentConfig

llm = OpenAILlm(
    model='o3-mini',

generate_content_config=OpenAIGenerateContentConfig(effort='high'),
)
```

Install the new OpenAI integration extra and use per-request api_key resolution in OpenAILlm.

```
$ pip install "google-adk[openai]"
```

Adds `ADK_ENABLE_SKILL_LIFECYCLE=1` environment variable to enable experimental skill lifecycle management, including an EPHEMERAL lifecycle (tools last one turn), an active-skill cap on `SkillToolset`, an opt-in `unload_skill` tool, and automatic pruning of unloaded skill instructions from later requests.

Adds `default_project_id` and `default_dataset_id` fields to `BigQueryToolConfig`.

Adds `RunConfig.service_tier` to expose and control the serving tier per invocation.

Adds `behavior` field to tool definitions to control Live non-blocking function call behavior.

Adds `context_builder` parameter to `RemoteA2aAgent` for customizing A2A request context.

+20 more

└──▷ **BREAKING ON UPGRADE**

- ! `BigQueryToolConfig` protected write mode now rejects multi-statement scripts, `CALL`, `EXPORT DATA`, and any other non-SELECT statement that the dry run does not place in the session's anonymous dataset.
- ! BigQuery protected mode now keeps its session in process memory instead of session state — temporary tables do not survive a restart or carry over to another replica.
- ! `OpenAIResponsesLlm` now ignores `thinking_config` (`thinking_level` / `thinking_budget`) with a warning; use `OpenAIGenerateContentConfig.effort` instead.
- ! `AgentEvaluator.evaluate` and `evaluate_eval_set` now raise `ValueError` when no eval cases are evaluated, instead of passing silently.
- ! Instruction templating now leaves `${var}` and backslash-escaped `\{var}` as written instead of filling them from state.
- ! Legacy live-audio modules under `google.adk.flows.llm_flows` (`audio_cache_manager`, `audio_transcriber`, `transcription_manager`) now emit a `DeprecationWarning` on import; test suites running with `-W error` will fail on these imports. Move to `google.adk.live`.

WAS THIS USEFUL? 👍 👎

Stacklok Mecatl

Feedback ⓘ v0.0.41 NOTES ↗ RANK

Mecatl is an open-source agent harness for running the same provider-agnostic agent loop locally, remotely, or on Kubernetes, with durable state, permissions, and gRPC/HTTP-SSE APIs.

> Mecatl v0.0.41 adds runtime policy enforcement via contextual guardrails, live MCP tool reconciliation, new Studio authorization controls and UI scaffolding, centralized provider-scoped model discovery, and several mecatui rendering upgrades.

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

01 **mecatui rendering and palette additions** NEW 60

Adds StatusML terminal hyperlink rendering, a bounded mention palette for scrollable, cursor-selectable @`-mentions`, and a bounded slash-command palette for scrollable, cursor-selectable slash commands, all in `mecatui`.

— Names the module and three concrete UI behaviours a reader can try. v0.0.41

THINNER COVERAGE BELOW

02 **MCP session-tool reconciliation and live refresh** NEW 50

Adds MCP source reconciliation and live session-tool refresh so running sessions pick up tool changes without restart.

— Clear before/after behaviour but no invocation detail given. v0.0.41

03 **Centralized provider-scoped model discovery** IMPROVED 48

Centralizes provider-scoped model discovery, replacing the alpha model-discovery contract.

— Gives before/after but no API or config names. v0.0.41

04 **Studio interface foundations** NEW 45

Adds a public-status indicator and a popup sign-in recovery flow, new settings routes and an About page, a workspace shell, and an appearance foundation (theming layer) to Studio.

— Names several distinct Studio additions but each described in one line only. v0.0.41

05 **Contextual guardrails in execution pipeline** NEW 43

Integrates contextual guardrails into the agent execution and approvals pipeline, enforcing policy checks at runtime.

— Describes mechanism but names no config key or flag. v0.0.41

06 **Authorization-review gate and plan control in Studio** NEW

40

Adds exact plan control and an authorization-review prerequisite gate to Studio.

— Named as a gate but no navigation or config specifics given. v0.0.41

07 **Provider request correlation for tracing** NEW

36

Correlates provider requests with the root session for end-to-end tracing.

— States purpose but no trace ID field or endpoint named. v0.0.41

v0.0.41 NOTES ↗ LATEST STABLE

Mecatl v0.0.41 adds contextual guardrails in the execution loop, MCP session-tool reconciliation, and Studio authorization-review controls.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.0.41 https://github.com/stacklok/mecatl.git
# already have the repo? check out this version:
$ git checkout v0.0.41
```

Integrates contextual guardrails into the agent execution and approvals pipeline, enforcing policy checks at runtime.

Adds MCP source reconciliation and live session-tool refresh so running sessions pick up tool changes without restart.

Adds exact plan control and an authorization-review prerequisite gate to Studio.

Adds a public-status indicator and popup sign-in recovery flow to Studio.

Adds settings routes and an About page to Studio.

+7 more

WAS THIS USEFUL? 👍 👎

Brig

Feedback ⓘ v0.3.0 NOTES ↗ RANK

Brig runs coding agents in a microVM sandbox, made up of the brig CLI and the brigd session daemon.

> Brig v0.3.0 adds out-of-the-box Linux support with a rootless install path, introduces guest port publishing and a raised 64 KiB secret cap, and ships a cluster of breaking changes to profile schema, network posture, `brig rm`, `brig shell`, and default guest identities.

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

01 Guest port publishing to host NEW

80

`brig network publish` exposes a guest port on the host, binding to `127.0.0.1`, letting a dev server running in a sandbox be reached from the local browser without opening the sandbox to the network.

Expose a dev server running in a sandbox to your local browser without opening the sandbox to the network.

```
$ brig network publish claude 3000
```

— Runnable command with concrete bind address given v0.3.0

02 brig rm now deletes guest home by default BREAKING

80

`brig rm` now deletes the guest home Brig created, including in-guest logins for codex, cursor, gemini, grok, opencode, and ubuntu. Pass `--home <dir>` to preserve a guest home across `brig rm`.

— Exact flag given to preserve prior behaviour v0.3.0

03 Network posture set only at sandbox creation BREAKING

75

A profile's `network:` key now sets the network posture only at sandbox creation time; changing it no longer re-applies to a sandbox that already has a recorded posture, including across stop/run cycles. Use the `--network` flag or `brig rm` to apply a new value.

— Explains before/after and gives workaround flag v0.3.0

04 Agents now boot as root with home moved to /root

BREAKING

70

claude-code, codex, gemini, grok, and opencode now boot `-stock:root` images and run as root in the guest, moving the guest home from `/home/<agent>` to `/root`. Any script, hook, or setting referencing the old absolute path must be updated.

— Names path change but no automated migration given v0.3.0

---

05 **Linux runtime bundle and rootless installs** NEW

65

`install.sh` now installs the full Linux runtime bundle — `nerdctl`, containerd, the `runc` shim, monitors, and guest kernel — automatically, with rootless support landing under `$HOME`.

— Names bundle components but no verification command given v0.3.0

---

06 **JSON output for brig version** NEW

65

Adds `--json` flag to `brig version` for machine-readable build output, for piping into scripts or CI checks that need structured build metadata.

Pipe `brig version` into scripts or CI checks that need structured build metadata.

```
$ brig version --json
```

— Exact flag and runnable example given, minimal mechanism v0.3.0

---

07 **Removal of hostCredential from profile schema** BREAKING

65

`hostCredential:` is removed from the profile schema — any profile file containing it will fail to parse. The `BRIG_ALLOW_EXPIRED` environment variable, which only applied to that field, is also removed.

— Names removed key and env var, no migration path given v0.3.0

THINNER COVERAGE BELOW

---

08 **Raised secret read cap to 64 KiB** IMPROVED

50

Raises the secret read cap to 64 KiB, with values sealed into a second keychain item.

— Names limit and mechanism but no invocable surface v0.3.0

---

09 **brig shell command removed** BREAKING

45

`brig shell` is removed. Use `brig sh` instead.

— Simple rename with clear replacement command v0.3.0

10 **Main and experimental Homebrew channels** NEW 30

Publishes a main and an experimental Homebrew channel for Brig.

— Bare announcement, no install command given v0.3.0

└──▷ **BREAKING ON UPGRADE**

! `hostCredential:` is removed from the profile schema — any profile file containing it will fail to parse. The `BRIG_ALLOW_EXPIRED` environment variable, which only applied to that field, is also removed.

! `brig rm` now deletes the guest home Brig created (including in-guest logins for codex, cursor, gemini, grok, opencode, and ubuntu). Pass `--home <dir>` to preserve a guest home across `brig rm`.

! A profile's `network:` key now sets the network posture only at sandbox creation time; changing it no longer re-applies to a sandbox that already has a recorded posture (including across stop/run cycles). Use `--network` flag or `brig rm` to apply a new value.

! `brig shell` is removed. Use `brig sh` instead.

! claude-code, codex, gemini, grok, and opencode now boot `-stock:root` images and run as root in the guest, moving the guest home from `/home/<agent>` to `/root`. Any script, hook, or setting referencing the old absolute path must be updated.

v0.3.0 NOTES ↗ LATEST STABLE

Brig v0.3.0 adds out-of-the-box Linux support with rootless installs, guest port publishing, and a 64 KiB secret cap.

└──▷ **GET THIS VERSION**

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

└──▷ **TRY IT**

Expose a dev server running in a sandbox to your local browser without opening the sandbox to the network.

```
$ brig network publish claude 3000
```

Pipe `brig version` into scripts or CI checks that need structured build metadata.

```
$ brig version --json
```

Adds `--json` flag to `brig version` for machine-readable build output.

Adds `brig network publish` to expose a guest port on the host (binds to `127.0.0.1`).

Raises the secret read cap to 64 KiB, with values sealed into a second keychain item.

`install.sh` now installs the full Linux runtime bundle (`nerdctl`, containerd, the `urunc` shim, monitors, and guest kernel) automatically, with rootless support landing under `$HOME`.

Publishes a main and an experimental Homebrew channel for Brig.

└──▷ **BREAKING ON UPGRADE**

- ! `hostCredential:` is removed from the profile schema — any profile file containing it will fail to parse. The `BRIG_ALLOW_EXPIRED` environment variable, which only applied to that field, is also removed.
- ! `brig rm` now deletes the guest home Brig created (including in-guest logins for codex, cursor, gemini, grok, opencode, and ubuntu). Pass `--home <dir>` to preserve a guest home across `brig rm`.
- ! A profile's `network:` key now sets the network posture only at sandbox creation time; changing it no longer re-applies to a sandbox that already has a recorded posture (including across stop/run cycles). Use `--network` flag or `brig rm` to apply a new value.
- ! `brig shell` is removed. Use `brig sh` instead.
- ! claude-code, codex, gemini, grok, and opencode now boot `-stock:root` images and run as root in the guest, moving the guest home from `/home/<agent>` to `/root`. Any script, hook, or setting referencing the old absolute path must be updated.

WAS THIS USEFUL? 👍 👎

LiveKit Agents

Feedback ⓘ livekit-agents@1.8.3 NOTES ↗

RANK

LiveKit Agents builds realtime voice and video AI agents on LiveKit's WebRTC infrastructure, with turn detection, interruption handling, and telephony.

> LiveKit Agents 1.8.3 expands provider and model coverage across its STT, TTS, and realtime plugins, moves the OpenAI Realtime inference client into the agents core library, and adds smaller telephony, workflow, and worker-logging improvements.

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

01 New provider and model support across STT/TTS plugins NEW

77

Adds `GPTLiveModel.with_azure` to the OpenAI plugin for connecting to Azure OpenAI realtime endpoints; adds `assemblyai/universal-3-6-pro` to the STT model type registry along with a new `language_confidence` field in AssemblyAI transcript metadata; adds a `phonic_model` option to select the Phonic model version; adds new Gemini 3.8 models to the Google TTS plugin, including expressive mode; adds `service_tier` configuration to the GPT Live model; adds configurable audio chunk duration for ElevenLabs realtime STT; adds support for realtime model literals in the inference client; and supports second-generation model naming in BitHuman plugin sessions.

Use the new AssemblyAI Universal-3 model via the inference STT client for high-accuracy transcription.

python

```python
from livekit.agents import inference, AgentSession

session = AgentSession(
    stt=inference.STT('assemblyai/universal-3-6-pro'),
    llm=inference.LLM('google/gemma-4-31b-it'),
    tts=inference.TTS('cartesia/sonic-3'),
)
```

— Names many providers/flags but each addition described in one line livekit-agents@1.8.3

THINNER COVERAGE BELOW

02 Worker process logging and agent naming IMPROVED

50

Adds `pid` to log lines emitted by the parent worker process, and reads the agent name from `livekit.toml` as a last fallback in the worker.

— Names config file and log field but limited mechanism detail livekit-agents@1.8.3

03 **verify_spelling added to GetEmailTask workflow** NEW

40

Adds `verify_spelling` to `GetEmailTask` in the workflows module.

— Names the field and task but no usage shown livekit-agents@1.8.3

04 **Tool results support reply_required** NEW

35

Adds support for `reply_required` in tool results.

— Names the field but no behavior or usage explained livekit-agents@1.8.3

05 **DTMF input resets user-away timer** IMPROVED

35

Resets the user-away timer on DTMF input.

— Explains behavior but not the mechanism or config livekit-agents@1.8.3

06 **OpenAI Realtime inference client moved into core** IMPROVED

27

Moves the OpenAI Realtime inference client into the agents core library.

— States the change with no migration detail or new interface livekit-agents@1.8.3

livekit-agents@1.8.3 NOTES ↗ LATEST STABLE

LiveKit Agents 1.8.3 adds Azure OpenAI realtime, AssemblyAI Universal-3, MistralAI chat completions, DTMF timer reset, and more plugin expansions.

└──▷ GET THIS VERSION

```
$ git clone --branch livekit-agents@1.8.3
https://github.com/livekit/agents.git
# already have the repo? check out this version:
$ git checkout livekit-agents@1.8.3
```

└──▷ USE IT

Use the new AssemblyAI Universal-3 model via the inference STT client for high-accuracy transcription.

```python
from livekit.agents import inference, AgentSession

session = AgentSession(
    stt=inference.STT('assemblyai/universal-3-6-pro'),
    llm=inference.LLM('google/gemma-4-31b-it'),
    tts=inference.TTS('cartesia/sonic-3'),
)
```

Adds `GPTLiveModel.with_azure` to the OpenAI plugin for connecting to Azure OpenAI realtime endpoints.

Adds `assemblyai/universal-3-6-pro` to the STT model type registry for use with the inference client.

Adds `language_confidence` field to AssemblyAI transcript metadata, surfaced via the STT plugin.

Adds `phonic_model` option to the Phonic plugin to select the Phonic model version.

Adds `verify_spelling` to `GetEmailTask` in the workflows module.

+10 more

WAS THIS USEFUL? 👍 👎

LangChain LangSmith

Feedback ⓘ announced · Sep 24, 2026

BLOG ↗ RANK

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

> LangSmith's headline addition this window is Engine v2, which introduces proactive Red Teaming for agents (private beta) alongside trajectory-inefficiency detection and production performance-trend tracking; alongside that it shipped a new `GET /v2/threads/stats` endpoint, tightened dataset-export permissions, and a batch of smaller platform, alerting, and UI fixes.

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

01 Red Teaming in LangSmith Engine NEW

75

Engine introduces Red Teaming, a proactive troubleshooting capability that analyzes production traces and repos to test agents for hallucinations and system-prompt violations before they surface in production (Private Beta for LangSmith Deployment users). Re-running a Red Team scan on a previously scanned project now starts from that project's Agent Overview and runs a shorter confirmation scan instead of re-deriving the application map.

— Explains mechanism, scope and before/after; no exact UI steps launch-20260924-ef06e4b3 changelog-20260926-d394ac7c

02 Accurate timeout status codes for tracing uploads IMPROVED

75

When a tracing batch upload times out, `/runs/multipart` and `/runs/batch` now return `408 Request Timeout` instead of `503`; a server-side timeout returns `504`.

— Named endpoints and exact status codes, no migration note changelog-20260926-d394ac7c

03 Expanded package support in code evaluators IMPROVED

70

The `system/code-evaluators:latest` snapshot now bundles pandas, jsonschema, SciPy, and scikit-learn alongside NumPy so evaluators can use ACE's supported Python packages without installing them.

— Named snapshot and packages, no usage command shown changelog-20260926-d394ac7c

THINNER COVERAGE BELOW

04 Performance trend tracking in Engine v2 NEW

50

Engine v2 tracks performance trends in error rate, latency, and cost across production traces, surfacing degradations in the same issue queue.

– Names metrics tracked, no threshold or UI path launch-20260924-ef06e4b3

05 **Engine detection of truncated tool-call results** NEW 50

Engine now reads the contents of successful tool results to file issues where a call returned less data than claimed or a value arrived truncated.

– Describes mechanism, no config or command given changelog-20260926-d394ac7c

06 **Alert text-filter matching independent of storage settings** IMPROVED 50

Alerts using input, output, or error text filters now match runs regardless of blob storage and ClickHouse search settings.

– Explains fix scope but no config example changelog-20260926-d394ac7c

07 **Trajectory inefficiency detection in Engine v2** NEW 45

Engine v2 detects inefficient agent trajectories — such as incorrect or repetitive tool calls — and proposes more efficient alternatives.

– States mechanism, no example or metric given launch-20260924-ef06e4b3

08 **Application tag assignment in Resource Tags** NEW 45

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

– Gives UI path, no further mechanism snapshot-20260926

09 **Persistent LangSmith Chat conversation state** NEW 40

LangSmith Chat keeps your conversation as you navigate and reload within a browser tab, starting a new one in a new tab or after a browser restart.

– Clear behavior description, no named surface changelog-20260926-d394ac7c

10 **Automatic free Serverless deployment assignment** IMPROVED 40

Your plan's included free Serverless deployment is now assigned automatically regardless of how it is created (web app, CLI, or API).

– States behavior, no config detail changelog-20260926-d394ac7c

11 **Cache Hit % shown in project stats** NEW 35

The project stats sidebar now shows Cache Hit % above streaming rate.

– Bare UI description, no further mechanism changelog-20260926-d394ac7c

12 **IAM role assumption for ECR** NEW 15

LangSmith adds IAM role assumption for ECR.

— Named only, no detail on configuration or scope snapshot-20260926

13 **Gateway fallback timeouts** NEW 15

LangSmith adds Gateway fallback timeouts.

— Named only, no configuration detail given snapshot-20260926

└──▷ **BREAKING ON UPGRADE**

! Scale-to-zero deployments block rollbacks to revisions running LangGraph API below version 0.13.0 or without a known compatible version; affected deployments must rebuild the target commit with a supported version.

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

> announced · Sep 24, 2026 BLOG ↗
>
> LangSmith Engine v2 adds red teaming, trajectory inefficiency detection, and pre-validated fix testing for agents.
>
> Introduces Red Teaming in LangSmith Engine, a proactive troubleshooting capability that analyzes production traces and repos to test agents for hallucinations and system-prompt violations before they surface in production (Private Beta for LangSmith Deployment users).
>
> Engine v2 detects inefficient agent trajectories — such as incorrect or repetitive tool calls — and proposes more efficient alternatives.
>
> Engine v2 tracks performance trends in error rate, latency, and cost across production traces, surfacing degradations in the same issue queue.

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Agentic Infrastructure

Google Gemini Enterprise Agent Platform

Feedback ⓘ changelog · Sep 26, 2026 CHANGELOG ↗ RANK

Gemini Enterprise Agent Platform provides Google's managed model training, inference, evaluation, and agent deployment services, formerly offered under Vertex AI.

> Gemini Enterprise Agent Platform expanded CodeMender's language coverage and reporting, introduced stateful streaming video generation through the Interactions API, and split its Python SDK into a new `google-cloud-agentplatform` package — a breaking change for existing agent code.

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

01 **CodeMender scanning for five new languages** NEW 65

CodeMender adds out-of-the-box vulnerability scanning support for C# (`.cs`), Rust (`.rs`), Kotlin (`.kt` , `.kts`), Ruby (`.rb`), and PHP (`.php`) to the default discovery configuration and initialization templates.

– Names exact file extensions and config areas but no CLI steps

changelog-20260926-9074dc93

THINNER COVERAGE BELOW

02 **Per-turn latency breakdowns in session exports** IMPROVED 50

Session and session export reporting now includes per-turn latency breakdowns, distinguishing model inference wait time from local tool execution time.

– Explains the metric split but no field names or export command

changelog-20260926-9074dc93

03 **CVP for Claude** NEW 10

CVP for Claude is added to the platform.

– Only a bare name is mentioned, no explanation of what it does

changelog-20260926-9074dc93

└──▷ **BREAKING ON UPGRADE**

! Agent Platform SDK for Python 2.0.1 decouples the agent surface from `google-cloud-aiplatform` into the new `google-cloud-agentplatform` package with restructured namespaces — existing code importing from `google-cloud-aiplatform` agent modules requires migration per the Agent Platform SDK for Python version 2.0.1 migration guide.

changelog · Sep 26, 2026 CHANGELOG ↗ LATEST STABLE

CodeMender gains interactive HTML reports and C#/Rust/Kotlin/Ruby/PHP scanning; Gemini 3.8 Flash Cyber, CVP for Claude, and stateful video generation arrive.

└──▷ **TRY IT**

Generate an interactive HTML security report and immediately open it in your browser after a CodeMender scan — useful for triage sessions where you want the visual dashboard and patch diffs without manually locating the output file.

```
$ cm report --format html --open
```

Start a stateful, streaming video generation interaction with Gemini Omni Flash so you can poll intermediate status over SSE and retrieve the completed video without re-sending the full prompt.

```
$ curl -X POST
https://interactions.agentplatform.googleapis.com/v1/interactions \
  -H 'Authorization: Bearer $(gcloud auth print-access-token)' \
  -H 'Content-Type: application/json' \
  -d '{"model": "gemini-omni-flash", "store": true, "stream": true,
"contents": [{"role": "user", "parts": [{"text": "Generate a 5-
second clip of a network topology diagram animating packet
flow."}]}]}'
```

Adds out-of-the-box vulnerability scanning support for C# (`.cs`), Rust (`.rs`), Kotlin (`.kt` , `.kts`), Ruby (`.rb`), and PHP (`.php`) to the default discovery configuration and initialization templates.

Enhances session and session exports to report per-turn latency breakdowns, distinguishing model inference wait time from local tool execution time.

└──▷ **BREAKING ON UPGRADE**

! Agent Platform SDK for Python 2.0.1 decouples the agent surface from `google-cloud-aiplatform` into the new `google-cloud-agentplatform` package with restructured namespaces — existing code importing from `google-cloud-aiplatform` agent modules requires migration per the Agent Platform SDK for Python version 2.0.1 migration guide.

WAS THIS USEFUL? 👍 👎

AWS Amazon Bedrock AgentCore

Feedback ⓘ

changelog · Sep 26, 2026 CHANGELOG ↗ RANK

Amazon Bedrock AgentCore provides managed agent runtimes, memory, identity, gateways, observability, and evaluation services.

> Amazon Bedrock AgentCore's harness gained an interactive WebSocket shell API for live session inspection and a new configuration field to route OpenAI model traffic through custom endpoints.

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

01 Interactive WebSocket shell into harness sessions NEW

83

The new `InvokeAgentRuntimeCommandShell` WebSocket API opens persistent, interactive shell sessions inside an AgentCore harness microVM, buffering up to 256 KB of output replay and supporting up to 10 independent concurrent shells per session.

Connect to an interactive shell inside a running AgentCore harness session to inspect environment state or run ad-hoc commands during agent execution.

```
$ wss://<agentcore-endpoint>/InvokeAgentRuntimeCommandShell
```

— Names endpoint, limits, and concurrency; lacks setup/auth detail

changelog-20260926-fe50cca1

changelog · Sep 26, 2026 CHANGELOG ↗ LATEST STABLE

AgentCore harness gains interactive WebSocket shells, lifecycle hooks, and custom OpenAI-compatible endpoint routing.

└──▷ **TRY IT**

Connect to an interactive shell inside a running AgentCore harness session to inspect environment state or run ad-hoc commands during agent execution.

```
$ wss://<agentcore-endpoint>/InvokeAgentRuntimeCommandShell
```

Route all OpenAI model calls through a corporate proxy or self-hosted compatible endpoint instead of hitting the OpenAI origin directly.

```json
{
  "modelConfig": {
    "provider": "openai",
    "model": "gpt-4o",
    "apiBase": "https://my-openai-proxy.example.com/v1"
  }
}
```

Adds `InvokeAgentRuntimeCommandShell` WebSocket API to open persistent, interactive shell sessions inside an AgentCore harness microVM, with up to 256 KB of buffered output replay and up to 10 independent concurrent shells per session.

Adds `apiBase` field to OpenAI model configurations in AgentCore harness, enabling routing of OpenAI requests through a custom gateway, proxy, self-hosted, or regional endpoint.

WAS THIS USEFUL? 👍 👎

Perplexity API

Feedback ⓘ snapshot-20260926 NOTES ↗ RANK

Perplexity API Platform - build with the Router, Agent, Search, and Embeddings APIs. Real-time, web-wide research and Q&A capabilities for your products.

> Perplexity API added Fast Search for lower-latency web queries, expanded the Agent API's model roster, made Fast Search the default for the Agent API's `fast` preset at a lower per-request cost, and announced the retirement of seven legacy OpenAI model IDs from the Agent and Router APIs.

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

01 **Retirement of legacy OpenAI model IDs** BREAKING 75

On October 24, 2026 at 00:00 UTC, the Agent API and Router API will retire `openai/gpt-5.4`, `openai/gpt-5.4-mini`, `openai/gpt-5.4-nano`, `openai/gpt-5.2`, `openai/gpt-5.1`, `openai/gpt-5`, and `openai/gpt-5-mini`; these IDs will no longer be accepted or returned by model-list endpoints after the cutoff, so direct model selections and fallback chains must be updated beforehand.

— Exact cutoff time and full list of retired model IDs given snapshot-20260926

02 **New model support in Agent API** NEW 60

The Agent API added support for `openai/gpt-6-sol`, `openai/gpt-6-luna`, `anthropic/claude-opus-5-5`, and `xai/grok-4.7`, letting workflows select these models directly via the `model` field.

Route Agent API agentic workflows through Claude Opus 5.5 for high-capability reasoning tasks without leaving the Perplexity platform.

```
$ curl -X POST https://api.perplexity.ai/agent \
  -H 'Authorization: Bearer <your_api_key>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "anthropic/claude-opus-5-5", "messages":
[{"role": "user", "content": "Summarize recent ransomware
campaigns targeting healthcare."}]}'
```

— Names all four model IDs with a working request example snapshot-20260926

└──▷ **BREAKING ON UPGRADE**

! On October 24, 2026 at 00:00 UTC, the Agent API and Router API will retire model IDs `openai/gpt-5.4`, `openai/gpt-5.4-mini`, `openai/gpt-5.4-nano`, `openai/gpt-5.2`, `openai/gpt-5.1`, `openai/gpt-5`, and `openai/gpt-5-mini` — these IDs will no longer be accepted or returned by model-list endpoints after the cutoff.

! On October 24, 2026 at 00:00 UTC, the Agent API and Router API will stop accepting the following model IDs — update all direct model selections and fallback chains before the cutoff: `openai/gpt-5.4`, `openai/gpt-5.4-mini`, `openai/gpt-5.4-nano`, `openai/gpt-5.2`, `openai/gpt-5.1`, `openai/gpt-5`, `openai/gpt-5-mini`. Retired IDs will no longer be returned by model-list endpoints after the cutoff.

snapshot-20260926 NOTES ↗ LATEST STABLE

Perplexity API: Adds `openai/gpt-6-sol` model support to the Agent API.

└──▷ **TRY IT**

Cut search latency by ~800 ms and reduce cost to $1.00/1,000 requests when running time-sensitive threat-intel lookups via the Search API.

```
$ curl -X POST https://api.perplexity.ai/search \
  -H 'Authorization: Bearer <your_api_key>' \
  -H 'Content-Type: application/json' \
  -d '{"query": "latest CVEs in Apache HTTP Server", "search_type":
"fast"}'
```

Route Agent API agentic workflows through Claude Opus 5.5 for high-capability reasoning tasks without leaving the Perplexity platform.

```
$ curl -X POST https://api.perplexity.ai/agent \
  -H 'Authorization: Bearer <your_api_key>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "anthropic/claude-opus-5-5", "messages": [{"role":
"user", "content": "Summarize recent ransomware campaigns targeting
healthcare."}]}'
```

Adds `openai/gpt-6-luna` model support to the Agent API.

Adds `anthropic/claude-opus-5-5` model support to the Agent API.

Adds `xai/grok-4.7` model support to the Agent API.

The Agent API `fast` preset now uses Fast Search (`search_type: "fast"`) by default, reducing `web_search` cost from $2.50 to $1.00 per 1,000 invocations.

└──▷ **BREAKING ON UPGRADE**

! On October 24, 2026 at 00:00 UTC, the Agent API and Router API will stop accepting the following model IDs — update all direct model selections and fallback chains before the cutoff: `openai/gpt-5.4`, `openai/gpt-5.4-mini`, `openai/gpt-5.4-nano`, `openai/gpt-5.2`, `openai/gpt-5.1`, `openai/gpt-`

`5`, `openai/gpt-5-mini`. Retired IDs will no longer be returned by model-list endpoints after the cutoff.

WAS THIS USEFUL? 👍 👎

Together AI

Feedback ⓘ changelog · Sep 26, 2026 CHANGELOG ↗

RANK

Run, train, and serve open-source AI models on Together AI.

> Together AI shipped TogetherLink for routing coding assistants through open models, plus canary/blue-green traffic rollouts, HIPAA-restricted placement, and idle-timeout autoscaling for dedicated endpoints, alongside FLUX image generation, prompt caching, OCR, and new client integrations for Hugging Face, Vercel AI SDK, and Mastra.

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

01 TogetherLink CLI for routing AI tools to open models NEW

95

Install via `curl -fsSL https://link.together.ai/install | bash`; run `togetherlink` for the interactive launcher or `togetherlink --main` to pin a session to a single model (e.g. `togetherlink --main claude together/Tev1-4B-experimental`). Routes Claude Code, Claude Desktop, Codex CLI, and ChatGPT Desktop through OSS models hosted on Together AI.

Pin a TogetherLink session to a specific Together AI-hosted model instead of using the auto router.

```
$ togetherlink --main claude together/Tev1-4B-experimental
```

Route Codex CLI through an OSS model on Together AI instead of OpenAI, using TogetherLink.

📍 TogetherLink

– Install command, flag, and target apps all named `snapshot-20260926`

`changelog-20260926-e38f5dd7`

02 Idle shutdown for dedicated deployments NEW

85

Adds `--inactive-timeout` CLI flag and `inactiveTimeout` management API field to scale a dedicated deployment to zero replicas after a configurable period of inactivity, stopping billing.

– Names flag and API field but gives no example call `changelog-20260926-e38f5dd7`

03 Traffic rollout strategies for dedicated endpoints NEW

85

Adds `tg beta endpoints rollout` CLI command and `client.beta.endpoints.rollouts` Python SDK surface to shift live traffic between deployments using canary, blue-green, or rolling strategies — without changing the endpoint URL.

— Named CLI command and SDK surface, no runnable example changelog-20260926-e38f5dd7

04 **HIPAA-attested cluster placement** NEW

80

Adds `--placement.hipaa` flag to the Together CLI and `compliance_policy` field in the Python SDK to restrict a dedicated deployment to HIPAA-attested clusters.

— Names flag and field, no worked example changelog-20260926-e38f5dd7

05 **Active-sessions autoscaling metric for TTS endpoints** NEW

80

Adds `active_sessions` as an autoscaling metric for TTS dedicated endpoints, targeting concurrent open client WebSocket sessions per replica (`AVERAGE_VALUE` only).

— Named metric and constraint, no config example given changelog-20260926-e38f5dd7

06 **Strict validation on dedicated endpoint management API**

80

BREAKING

The dedicated endpoint management API now rejects unknown JSON request-body fields and unknown query parameters with an HTTP error naming the offending field, instead of silently ignoring it; clients sending retired or misspelled keys (e.g. `inactive_timeout` instead of `inactiveTimeout`) must remove them.

— Names exact field rename and new error behaviour changelog-20260926-e38f5dd7

07 **FLUX image generation and editing models** NEW

65

Adds FLUX model family support including FLUX.2 and FLUX.1 Kontext for image generation and editing, and adds image generation via Flux2 deployed on Together's managed GPU infrastructure using dedicated containers.

— Names model variants but no usage example given snapshot-20260926

08 **Deployment profiles replace manual hardware controls** BREAKING

60

The dedicated endpoint create form in the console now shows Deployment profiles choice cards that pin hardware, quantization, and decoding options from certified configs, replacing the previously separate Quantization, Hardware, LoRA, and speculative-decoding controls.

— UI change described, no exact navigation path changelog-20260926-e38f5dd7

09 **Client library and framework integrations** NEW 60

Adds a Hugging Face Inference client integration to call Together models, a Vercel AI SDK integration for calling Together models from TypeScript, React, or Next.js, and a Mastra agent framework integration to point agents at Together AI serverless models.

— Names three integrations but no code sample product docs

THINNER COVERAGE BELOW

10 **Prompt caching for reused prefixes** NEW 45

Adds prompt caching to reuse previously processed prompt prefixes, reducing input cost and time to first token.

— Explains benefit, no config or API named snapshot-20260926

11 **OpenAI-compatible endpoint for custom models** NEW 45

Adds the ability to serve a custom model behind an OpenAI-compatible endpoint using dedicated containers.

— States capability, no endpoint path or config given snapshot-20260926

12 **GPT-OSS 120B model access** NEW 25

Adds access to the GPT-OSS 120B open-weight reasoning model.

— Mentioned only in summary, no further detail product docs

13 **OCR text extraction from images** NEW 20

Adds OCR capability to pull text out of images.

— Bare description, no model or API named snapshot-20260926

└──▷ **BREAKING ON UPGRADE**

! The dedicated endpoint management API now rejects unknown JSON request-body fields and unknown query parameters with an HTTP error (previously silently ignored); clients sending retired or misspelled keys (e.g. `inactive_timeout` instead of `inactiveTimeout`) must remove them.

! The console create-endpoint and new-deployment forms no longer show separate Quantization, Hardware, LoRA, and speculative-decoding controls; these are replaced by Deployment profiles choice cards.

changelog · Sep 26, 2026 CHANGELOG ↗ STABLE

Together AI adds TogetherLink beta, traffic rollouts, idle shutdown, HIPAA placement, and active_sessions autoscaling for TTS deployments.

└──▷ **TRY IT**

Pin a TogetherLink session to a specific Together AI-hosted model instead of using the auto router.

```
$ togetherlink --main claude together/Tev1-4B-experimental
```

Installs `togetherlink` via `curl -fsSL https://link.together.ai/install | bash`; run `togetherlink` for the interactive launcher or `togetherlink --main` to pin a session to a single model.

Adds `--inactive-timeout` CLI flag and `inactiveTimeout` management API field to scale a dedicated deployment to zero replicas after a configurable period of inactivity, stopping billing.

Adds `tg beta endpoints rollout` CLI command and `client.beta.endpoints.rollouts` Python SDK surface to shift live traffic between deployments using canary, blue-green, or rolling strategies — without changing the endpoint URL.

Adds `--placement.hipaa` flag to the Together CLI and `compliance_policy` field in the Python SDK to restrict a dedicated deployment to HIPAA-attested clusters.

Adds `active_sessions` as an autoscaling metric for TTS dedicated endpoints, targeting concurrent open client WebSocket sessions per replica (`AVERAGE_VALUE` only).

+2 more

└──▷ **BREAKING ON UPGRADE**

- ! The dedicated endpoint management API now rejects unknown JSON request-body fields and unknown query parameters with an HTTP error (previously silently ignored); clients sending retired or misspelled keys (e.g. `inactive_timeout` instead of `inactiveTimeout`) must remove them.
- ! The console create-endpoint and new-deployment forms no longer show separate Quantization, Hardware, LoRA, and speculative-decoding controls; these are replaced by Deployment profiles choice cards.

snapshot-20260926 NOTES ↗ STABLE

Together AI adds TogetherLink for OSS model routing, prompt caching, Flux2/Wan 2.1 generation, and OpenAI-compatible dedicated endpoints.

└──▷ **HOW TO FIND IT**

Route Codex CLI through an OSS model on Together AI instead of OpenAI, using TogetherLink.

📍 TogetherLink

Adds `TogetherLink` to route Claude Code, Claude Desktop, Codex CLI, and ChatGPT Desktop through OSS models hosted on Together AI.

Adds prompt caching to reuse previously processed prompt prefixes, reducing input cost and time to first token.

Adds image generation support via Flux2 deployed on Together's managed GPU infrastructure using dedicated containers.

Adds the ability to serve a custom model behind an OpenAI-compatible endpoint using dedicated containers.

Adds OCR capability to pull text out of images.

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.8.78 NOTES ↗ RANK

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

> HyperFrames v0.8.78 adds a project history command for auditing and reverting changes, alongside new shader transition and player APIs.

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

01 **Project history command for change management** NEW

72

The new `hyperframes history` command lets users list, undo, and restore a project's recorded changes.

— Names exact command and its three operations. v0.8.78

THINNER COVERAGE BELOW

02 **Seam transition loop preview in shader API** NEW

55

The shader transitions API adds `playSeamTransitionLoop` to preview a seam transition on a loop.

— Names the exact API method but not full mechanism. v0.8.78

03 **Player picture size, duration event, and click-to-play opt-out**

NEW

48

The player now exposes picture size, a `durationchange` event, and an option to opt out of click-to-play behavior.

— Names three player surfaces without usage detail. v0.8.78

v0.8.78 NOTES ↗ LATEST STABLE

HyperFrames v0.8.78 adds `hyperframes history` for project change management and new player/shader APIs.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.8.78 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.78
```

> Adds `hyperframes history` command to list, undo, and restore a project's recorded changes.
>
> Adds `playSeamTransitionLoop` to the shader transitions API to preview a seam on a loop.
>
> Exposes picture size, `durationchange` event, and a click-to-play opt-out on the player.

WAS THIS USEFUL?  

RunPod

Feedback ⓘ announced · Sep 25, 2026 BLOG ↗ RANK

AI infrastructure with on-demand GPUs and serverless compute. Run training, inference, and batch workloads on the cloud with Runpod.

> RunPod's runpod-python 0.10.0 adds a local HTTP API server for interactively testing serverless worker handlers, replacing the static test-input workflow.

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

01 **Local API server for testing worker handlers** NEW 88

The `--rp_serve_api` flag launches a local API server on port 8000 for interactive testing of serverless worker handlers, replacing the static `test_input.json` / `--test_input` workflow. The `--rp_api_host='0.0.0.0'` argument binds the server to all interfaces, enabling remote access when hosted on a standard RunPod GPU pod, and auto-generated API documentation is exposed at `http://localhost:8000/docs` while the server is running.

Spin up a local API server to interactively test your serverless worker handler before deploying, instead of relying on a static test_input.json file.

```
$ python whatever.py --rp_serve_api
```

Test GPU-dependent workers on a RunPod standard pod by binding the API server to all interfaces so you can submit requests remotely.

```
$ python whatever.py --rp_serve_api --rp_api_host='0.0.0.0'
```

— Names exact flags, port, docs URL and runnable commands. launch-20260925-8f1f4fb6

announced · Sep 25, 2026 BLOG ↗

runpod-python 0.10.0 adds `--rp`_serve_api to launch a local HTTP API server for testing serverless worker handlers

└──▷ **TRY IT**

Spin up a local API server to interactively test your serverless worker handler before deploying, instead of relying on a static test_input.json file.

```
$ python whatever.py --rp_serve_api
```

Test GPU-dependent workers on a RunPod standard pod by binding the API server to all interfaces so you can submit requests remotely.

```
$ python whatever.py --rp_serve_api --rp_api_host='0.0.0.0'
```

Adds `--rp_serve_api` flag to launch a local API server on port 8000 for interactive testing of serverless worker handlers, replacing the static `test_input.json` / `--test_input` workflow.

Adds `--rp_api_host='0.0.0.0'` argument to bind the local API server to all interfaces, enabling remote access when hosted on a standard RunPod GPU pod.

Exposes auto-generated API documentation at `http://localhost:8000/docs` when the local server is running.

WAS THIS USEFUL? 👍 👎

KServe

Feedback ⓘ v0.21.0 NOTES ↗ RANK

KServe is a Kubernetes-native inference platform serving predictive and generative models behind a standard interface, with autoscaling and canary rollouts.

> KServe v0.21.0 adds Dynamic Resource Allocation support for GPU scheduling, two new model storage URI schemes, and richer autoscaling and rollout controls for LLMInferenceService, alongside canary traffic splitting for raw deployments and TLS/mTLS hardening.

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

01 DRA resource claims in ServingRuntime NEW

85

Adds `resourceClaims` support to `ServingRuntime` for Dynamic Resource Allocation (DRA), enabling GPU and accelerator resource scheduling via Kubernetes DRA. A `ServingRuntime` can reference a `resourceClaimTemplateName` under `resourceClaims` to request a GPU through a resource claim template.

Use DRA resource claims in a ServingRuntime to schedule a GPU via Dynamic Resource Allocation.

yaml

```yaml
apiVersion: serving.kserve.io/v1alpha1
kind: ServingRuntime
metadata:
  name: vllm-dra
spec:
  resourceClaims:
    - name: gpu
      resourceClaimTemplateName: my-gpu-claim-template
```

— Runnable YAML example with named field, but no scheduling scope details. v0.21.0

02 KEDA autoscaling and rollout controls for LLMInferenceService

NEW

75

`LLMInferenceService` gains `idleReplicaCount=0` support for KEDA autoscaling, enabling true scale-to-zero (`autoscaling: idleReplicaCount: 0`), a `rolloutStrategy` field to control rolling update behavior, and direct KEDA scaling without requiring an intermediate component.

Enable true scale-to-zero for an LLMInferenceService by setting `idleReplicaCount` to 0 in the KEDA autoscaling config.

```yaml
autoscaling:
  idleReplicaCount: 0
```

— Named fields with one runnable config example. v0.21.0

03 **New model storage URI schemes** NEW 65

Adds the `oci+fetch://` URI scheme for KServe-side OCI image pull as a new model delivery path alongside existing OCI approaches, and the `ms://` URI scheme for ModelScope storage download, letting practitioners pull models directly from ModelScope (e.g. `storageUri: ms://Qwen/Qwen2.5-7B-Instruct`).

Pull a model from ModelScope in an InferenceService spec instead of HuggingFace Hub, using the new `ms://` URI scheme.

```yaml
storageUri: ms://Qwen/Qwen2.5-7B-Instruct
```

— Two named schemes, one with a runnable example. v0.21.0

THINNER COVERAGE BELOW

04 **Canary traffic splitting for RawDeployment HTTPRoutes**

IMPROVED 40

Adds canary traffic splitting to RawDeployment HTTPRoutes, extending canary support to the raw Kubernetes deployment mode.

— No example or configuration shown for enabling it. v0.21.0

05 **Cluster TLS integration and mTLS-aware transformer calls** NEW

40

Adds cluster TLS security profile integration, aligning KServe controllers with cluster-wide TLS settings, and CA bundle environment variable injection for `httpx` in the transformer, enabling mTLS-aware HTTP calls from transformers.

— No exact env var name or config path given. v0.21.0

06 **vLLM render deployment replaces UDS tokenizer sidecar**

IMPROVED 40

Replaces the UDS tokenizer sidecar with a vLLM render deployment for `LLMInferenceService`, changing how tokenizer sidecars are provisioned.

— Describes the change but gives no migration steps. v0.21.0

07 **WVA annotation-based discovery migration** BREAKING 40

Migrates the Workload Vertical Autoscaler (WVA) from a VA CRD to annotation-based discovery in `LLMInferenceService`.

— No migration path or affected config shown. v0.21.0

08 **Restructured CRD management for component installation**

IMPROVED 30

Restructures CRD management to support independent installation of KServe components.

— Vague on what independent installation entails. v0.21.0

09 **Python 3.13 support** NEW 20

Adds Python 3.13 support across KServe Python packages.

— Bare statement of version support, no further detail. v0.21.0

v0.21.0 NOTES ↗ LATEST STABLE

KServe v0.21.0 adds DRA resource claims, ModelScope storage, OCI image pull, KEDA scale-to-zero, canary traffic splitting, and Python 3.13 support.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.21.0 https://github.com/kserve/kserve.git
# already have the repo? check out this version:
$ git checkout v0.21.0
```

└──▷ **USE IT**

Pull a model from ModelScope in an InferenceService spec instead of HuggingFace Hub, using the new `ms://` URI scheme.

```yaml
storageUri: ms://Qwen/Qwen2.5-7B-Instruct
```

Enable true scale-to-zero for an LLMInferenceService by setting `idleReplicaCount` to 0 in the KEDA autoscaling config.

```yaml
autoscaling:
  idleReplicaCount: 0
```

Use DRA resource claims in a ServingRuntime to schedule a GPU via Dynamic Resource Allocation.

```yaml
apiVersion: serving.kserve.io/v1alpha1
kind: ServingRuntime
metadata:
  name: vllm-dra
spec:
  resourceClaims:
    - name: gpu
      resourceClaimTemplateName: my-gpu-claim-template
```

Adds `resourceClaims` support to `ServingRuntime` for Dynamic Resource Allocation (DRA), enabling GPU and accelerator resource scheduling via Kubernetes DRA.

Adds `oci+fetch://` URI scheme for KServe-side OCI image pull, providing a new model delivery path alongside existing OCI approaches.

Adds `ms://` URI scheme for ModelScope storage download support, letting practitioners pull models directly from ModelScope.

Adds `idleReplicaCount=0` support for `LLMInferenceService` KEDA autoscaling, enabling true scale-to-zero behavior.

Adds `rolloutStrategy` field to `LLMInferenceService` for controlling rolling update behavior.

+8 more

WAS THIS USEFUL? 👍 👎

OpenRouter

Feedback ⓘ snapshot-20260926 NOTES ↗ RANK

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

> OpenRouter added asynchronous batch inference, letting users submit large workloads in a single request and receive results within 24 hours at roughly half the standard per-token price.

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

01 **Async batch inference API** NEW 63

Submit a full workload in a single `POST` request and receive results within 24 hours, priced at roughly half the standard per-token rate.

— Names request method and pricing/timing but no endpoint path snapshot-20260926

snapshot-20260926 NOTES ↗ STABLE

> OpenRouter adds async batch inference — one POST, results within 24 hours, at roughly half the per-token price.

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

MLX Serve

Feedback ⓘ v26.9.6 NOTES ↗ RANK

The mlx-serve runtime provides Apple Silicon inference with OpenAI- and Anthropic-compatible APIs, chat, agent mode, and tool calling.

> mlx-serve v26.9.6 adds a new decision-scoring API, tunable speculative-decoding lookups, mid-run agent steering, and several M5 Ultra kernel-fusion speedups.

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

01 Decisions endpoint for game/app state NEW

90

Adds `POST /v1/decisions` endpoint to score, answer yes/no, and resolve choice questions about game or app state using Laya checkpoints (`aac6fef/laya-multilingual-mlx`), batching concurrent requests into one pass.

Query the new decisions endpoint to get a scored yes/no answer about current game state using a Laya checkpoint.

```
$ curl -X POST http://localhost:8080/v1/decisions \
  -H 'Content-Type: application/json' \
  -d '{"model": "aac6fef/laya-multilingual-mlx", "question": "Is the player in danger?", "state": {"health": 5, "enemies_nearby": 3}}'
```

— Named endpoint, model checkpoint, and runnable curl example given. v26.9.6

02 MTP draft-from-conversation lookup tuning NEW

90

Adds `MLX_SERVE_MTP_LOOKUP=0` environment variable to disable the new MTP draft-from-conversation lookup, which is on by default and accelerates file rewrites and edits 1.3–1.5x by drafting from the conversation itself when the reply copies it.

Disable MTP draft-from-conversation lookup when you want deterministic decode behaviour and do not need the 1.3–1.5x speedup on file-rewrite workloads.

```
$ MLX_SERVE_MTP_LOOKUP=0 mlx-serve --mtp
```

— Names the exact env var, default state, and measured speedup. v26.9.6

03 Thinking mode control via chat_template_kwargs NEW

80

Adds `chat_template_kwargs: {"preserve_thinking": true}` in Model Settings to restore full thinking replay for Qwen3 agents, which is suppressed by default in this release to halve context usage in long sessions; also supports `chat_template_kwargs` in per-request payloads (e.g. `{"enable_thinking": false}`) to control thinking mode at the request level.

— Names exact config keys and values but no runnable example. v26.9.6

04 **mlx-serve pull for TTS assets** NEW 75

Adds `mlx-serve pull` command to download all assets required by Kokoro and Qwen3-TTS for text-to-speech.

Pre-download all speech assets so Kokoro and Qwen3-TTS can speak without a runtime fetch.

```
$ mlx-serve pull
```

— Runnable command shown with named models it fetches. v26.9.6

05 **Kernel fusion throughput gains** IMPROVED 70

Fuses GDN layers into a single GPU dispatch for a few percent throughput gain on every supported Mac including 27B models; fuses MoE verification across 2 or more streams on M5 Ultra chips (+13% at 4 streams) and runs Ultra chip verification through the fused MoE kernel (+14%).

— Concrete performance numbers but automatic, no user action. v26.9.6

THINNER COVERAGE BELOW

06 **Full-conversation image delivery** IMPROVED 55

Delivers every image in a conversation to the model, including earlier turns and tool-returned images (e.g. from Claude Code's Read), without re-encoding already-seen images on subsequent turns.

— Explains mechanism and scope but no config to invoke it. v26.9.6

07 **Logprobs with structured response formats** IMPROVED 55

Surfaces `logprobs` support when used together with `response_format` (JSON mode and JSON schema).

— Names the fields but no example of combined use. v26.9.6

08 **Mid-run agent steering** NEW 50

Enables mid-run agent steering: pressing Return while an agent is running queues a note that the agent picks up at its next tool round.

— Describes behaviour but no config or command surface. v26.9.6

09 **App renamed to MLX-Serve** BREAKING 40

Renames the app and download artifact to MLX-Serve.app / MLX-Serve.dmg; settings carry over automatically.

— Simple rename note, minimal mechanism described. v26.9.6

└──▷ ALSO FROM THESE RELEASES

v26.9.6 NOTES ↗ LATEST STABLE

mlx-serve v26.9.6 adds POST `/v1`/decisions, MTP lookup tuning, M5 Ultra fused-kernel speedups, and mid-run agent steering

└──▷ GET THIS VERSION

```
$ git clone --branch v26.9.6 https://github.com/ddalcu/mlx-serve.git
# already have the repo? check out this version:
$ git checkout v26.9.6
```

└──▷ TRY IT

Query the new decisions endpoint to get a scored yes/no answer about current game state using a Laya checkpoint.

```
$ curl -X POST http://localhost:8080/v1/decisions \
  -H 'Content-Type: application/json' \
  -d '{"model": "aac6fef/laya-multilingual-mlx", "question": "Is the player in danger?", "state": {"health": 5, "enemies_nearby": 3}}'
```

Disable MTP draft-from-conversation lookup when you want deterministic decode behaviour and do not need the 1.3–1.5x speedup on file-rewrite workloads.

```
$ MLX_SERVE_MTP_LOOKUP=0 mlx-serve --mtp
```

Pre-download all speech assets so Kokoro and Qwen3-TTS can speak without a runtime fetch.

```
$ mlx-serve pull
```

Adds `POST /v1/decisions` endpoint to score, answer yes/no, and resolve choice questions about game or app state using Laya checkpoints (`aac6fef/laya-multilingual-mlx`), batching concurrent requests into one pass.

Adds `MLX_SERVE_MTP_LOOKUP=0` environment variable to disable the new MTP draft-from-conversation lookup (on by default), which accelerates file rewrites and edits 1.3–1.5x by drafting from the conversation itself when the reply copies it.

Adds `chat_template_kwargs: {"preserve_thinking": true}` in Model Settings to restore full thinking replay for Qwen3 agents (suppressed by default in this release to halve context usage in long sessions).

Supports `chat_template_kwargs` in per-request payloads (e.g. `{"enable_thinking": false}`) to control thinking mode at the request level.

Enables mid-run agent steering: pressing Return while an agent is running queues a note that the agent picks up at its next tool round.

+6 more

WAS THIS USEFUL? 👍 👎

KoboldCpp

Feedback ⓘ v1.122 NOTES ↗ CODE ↗ RANK

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

> KoboldCpp v1.122 introduces a built-in agentic coding harness with MCP tool support, alongside independent ubatch sizing, a new autoswap threshold flag, music-mode audio generation, structured streaming error responses, and a build-portability breaking change.

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

01 Integrated coding agent with MCP tools NEW 95

Adds `--agent` launch flag (also togglable from the Admin tab) enabling the built-in KoboldCpp Agent, a lightweight agentic harness compatible with any OpenAI Chat Completions endpoint, with 9 built-in tools, support for `AGENTS.md`, context compaction, and three tool-call approval modes (`on` / `auto` / `off`); it requires at least 28k context and 8k generation amount. The agent can be extended with additional tools by loading an `mcp.json` file, where MCP tools execute server-side while Agent tools execute on the agent client.

Launch KoboldCpp with the integrated agent enabled for a local coding-assistant session, pointing it at a model with sufficient context.

```
$ koboldcpp-linux-x64 --model my-model.gguf --contextsize 32768 --genamt 8192 --agent
```

Extend the KoboldCpp Agent with a custom MCP server so the agent can call your own tools at runtime.

```json
{
  "mcpServers": {
    "demo": {
      "url": "http://localhost:8000/mcp"
    }
  }
}
```

— Full flag, requirements, config file and modes all named v1.122

02 Portable build environment variable requirement BREAKING

75

`koboldcpp.sh` now optimizes builds for the local platform by default; package maintainers producing portable builds must set the `KCPP_PORTABLE=1` environment variable, or the resulting binary may not run on systems other than the build host.

— Exact env var and failure consequence given v1.122

---

03 **Autoswap threshold flag** NEW 70

Adds `--autoswapthreshold` flag to set the model size limit at which autoswap triggers, preventing unnecessary swaps for small models.

Prevent autoswap from triggering on small models loaded alongside a larger one by capping the swap threshold.

```
$ koboldcpp-linux-x64 --model my-model.gguf --autoswapthreshold
4096
```

— Exact flag with a runnable command example v1.122

---

04 **Independent ubatch size configuration** NEW 65

Adds independent `ubatchsize` configuration (and a corresponding `ubatch` slider in the GUI), separate from `batchsize`, mirroring llama.cpp behavior; pipeline-parallel mode now auto-enables when `ubatch` is smaller than `batchsize`.

— Names config and behavior but no runnable example v1.122

THINNER COVERAGE BELOW

---

05 **Structured errors for streaming failures** IMPROVED 55

Generation failures during streaming now return a structured error object (HTTP 500 before headers, or an SSE error after streaming starts) instead of silently failing.

— Explains error format but no endpoint or code sample v1.122

---

06 **Audio code generation in music mode** NEW 30

Supports audio code generation when using the LLM in music/lyrics mode via a two-step process, with an improved music UI.

— Thin description with no mechanism or UI path v1.122

└──▷ **BREAKING ON UPGRADE**

! `koboldcpp.sh` now optimizes builds for the local platform by default; package maintainers producing portable builds must set the `KCPP_PORTABLE=1` environment variable, or the resulting binary may not run on systems other than the build host.

v1.122 NOTES ↗ CODE ↗ LATEST STABLE

KoboldCpp v1.122 ships a built-in agentic coding harness with MCP tool support, plus independent `ubatchsize` control and a new `--autoswapthreshold` flag.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v1.122
https://github.com/LostRuins/koboldcpp.git
# already have the repo? check out this version:
$ git checkout v1.122
```

└──▷ **TRY IT**

Launch KoboldCpp with the integrated agent enabled for a local coding-assistant session, pointing it at a model with sufficient context.

```
$ koboldcpp-linux-x64 --model my-model.gguf --contextsize 32768 --
genamt 8192 --agent
```

Prevent autoswap from triggering on small models loaded alongside a larger one by capping the swap threshold.

```
$ koboldcpp-linux-x64 --model my-model.gguf --autoswapthreshold 4096
```

Extend the KoboldCpp Agent with a custom MCP server so the agent can call your own tools at runtime.

mcp.json

```
{
  "mcpServers": {
    "demo": {
      "url": "http://localhost:8000/mcp"
    }
  }
}
```

Adds `--agent` launch flag (also togglable from the Admin tab) to enable the integrated KoboldCpp Agent, a lightweight agentic harness compatible with any OpenAI Chat Completions endpoint, with 9 built-in tools, support for `AGENTS.md`, context compaction, and three tool-call approval modes (`on` / `auto` / `off`); requires at least 28k ctx and 8k gen amount.

Adds `--autoswapthreshold` flag to set the model size limit at which autoswap

triggers, preventing unnecessary swaps for small models.

Supports loading a `mcp.json` file to extend the KoboldCpp Agent with additional MCP tools; MCP tools execute server-side while Agent tools execute on the agent client.

Adds independent `ubatchsize` configuration (and a corresponding `ubatch` slider in the GUI), separate from `batchsize`, mirroring llama.cpp behavior; pipeline-parallel mode now auto-enables when `ubatch` is smaller than `batchsize`.

Supports audio code generation when using the LLM in music/lyrics mode (two-step process), with an improved music UI.

+1 more

└─▷ **BREAKING ON UPGRADE**

! `koboldcpp.sh` now optimizes builds for the local platform by default; package maintainers producing portable builds must set the `KCPP_PORTABLE=1` environment variable, or the resulting binary may not run on systems other than the build host.

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ changelog · Sep 26, 2026

CHANGELOG ↗ RANK

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

> LangSmith's headline addition this window is Engine v2, which introduces proactive Red Teaming for agents (private beta) alongside trajectory-inefficiency detection and production performance-trend tracking; alongside that it shipped a new `GET /v2/threads/stats` endpoint, tightened dataset-export permissions, and a batch of smaller platform, alerting, and UI fixes.

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

01 **Red Teaming in LangSmith Engine** NEW 75

Engine introduces Red Teaming, a proactive troubleshooting capability that analyzes production traces and repos to test agents for hallucinations and system-prompt violations before they surface in production (Private Beta for LangSmith Deployment users). Re-running a Red Team scan on a previously scanned project now starts from that project's Agent Overview and runs a shorter confirmation scan instead of re-deriving the application map.

– Explains mechanism, scope and before/after; no exact UI steps launch-20260924-ef06e4b3 changelog-20260926-d394ac7c

02 **Accurate timeout status codes for tracing uploads** IMPROVED 75

When a tracing batch upload times out, `/runs/multipart` and `/runs/batch` now return `408 Request Timeout` instead of `503`; a server-side timeout returns `504`.

– Named endpoints and exact status codes, no migration note changelog-20260926-d394ac7c

03 **Expanded package support in code evaluators** IMPROVED 70

The `system/code-evaluators:latest` snapshot now bundles pandas, jsonschema, SciPy, and scikit-learn alongside NumPy so evaluators can use ACE's supported Python packages without installing them.

– Named snapshot and packages, no usage command shown changelog-20260926-d394ac7c

THINNER COVERAGE BELOW

04 **Performance trend tracking in Engine v2** NEW 50

Engine v2 tracks performance trends in error rate, latency, and cost across production traces, surfacing degradations in the same issue queue.

– Names metrics tracked, no threshold or UI path `launch-20260924-ef06e4b3`

05 **Engine detection of truncated tool-call results** NEW 50

Engine now reads the contents of successful tool results to file issues where a call returned less data than claimed or a value arrived truncated.

– Describes mechanism, no config or command given `changelog-20260926-d394ac7c`

06 **Alert text-filter matching independent of storage settings** IMPROVED 50

Alerts using input, output, or error text filters now match runs regardless of blob storage and ClickHouse search settings.

– Explains fix scope but no config example `changelog-20260926-d394ac7c`

07 **Trajectory inefficiency detection in Engine v2** NEW 45

Engine v2 detects inefficient agent trajectories — such as incorrect or repetitive tool calls — and proposes more efficient alternatives.

– States mechanism, no example or metric given `launch-20260924-ef06e4b3`

08 **Application tag assignment in Resource Tags** NEW 45

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

– Gives UI path, no further mechanism `snapshot-20260926`

09 **Persistent LangSmith Chat conversation state** NEW 40

LangSmith Chat keeps your conversation as you navigate and reload within a browser tab, starting a new one in a new tab or after a browser restart.

– Clear behavior description, no named surface `changelog-20260926-d394ac7c`

10 **Automatic free Serverless deployment assignment** IMPROVED 40

Your plan's included free Serverless deployment is now assigned automatically regardless of how it is created (web app, CLI, or API).

– States behavior, no config detail `changelog-20260926-d394ac7c`

11 **Cache Hit % shown in project stats** NEW 35

The project stats sidebar now shows Cache Hit % above streaming rate.

— Bare UI description, no further mechanism changelog-20260926-d394ac7c

12 **IAM role assumption for ECR** NEW 15

LangSmith adds IAM role assumption for ECR.

— Named only, no detail on configuration or scope snapshot-20260926

13 **Gateway fallback timeouts** NEW 15

LangSmith adds Gateway fallback timeouts.

— Named only, no configuration detail given snapshot-20260926

└──▷ BREAKING ON UPGRADE

! Scale-to-zero deployments block rollbacks to revisions running LangGraph API below version 0.13.0 or without a known compatible version; affected deployments must rebuild the target commit with a supported version.

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

changelog · Sep 26, 2026 CHANGELOG ↗ LATEST STABLE

LangSmith: The `system/code-evaluators:latest` snapshot bundles pandas, jsonschema, SciPy, and scikit-learn alongside NumPy so evaluators can use ACE's supported Python packages without installing them.

The project stats sidebar now shows Cache Hit % above streaming rate.

Alerts using input, output, or error text filters now match runs regardless of blob storage and ClickHouse search settings.

Engine now reads the contents of successful tool results to file issues where a call returned less data than claimed or a value arrived truncated.

An Engine Red Team run on a previously scanned project now starts from that project's Agent Overview and runs a shorter confirmation scan instead of re-deriving the application map.

LangSmith Chat keeps your conversation as you navigate and reload within a browser tab, starting a new one in a new tab or after a browser restart.

+2 more

└──▷ BREAKING ON UPGRADE

! Scale-to-zero deployments block rollbacks to revisions running LangGraph API below version 0.13.0 or without a known compatible version; affected deployments must rebuild the target commit with a supported version.

snapshot-20260926 NOTES ↗ STABLE

LangSmith adds `/v2` /threads/stats endpoint, IAM role assumption for ECR, Gateway fallback timeouts, and OAuth session owner scoping

└──▷ **USE IT**

Set the license-expiration warning window to 14 days so operators get earlier notice before a self-hosted LangSmith license expires.

.env

```
LICENSE_EXPIRATION_WARNING_DAYS=14
```

Adds `GET /v2/threads/stats` endpoint to retrieve thread and trace counts, latency, token, cost, and feedback statistics for a tracing project, including on self-hosted ClickHouse-backed deployments.

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

└──▷ **BREAKING ON UPGRADE**

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

WAS THIS USEFUL? 👍 👎

UK AI Security Institute Inspect AI

Feedback ⓘ

0.3.269 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 added a `litellm-proxy` model provider and DeepSeek-V4.1-Flash support, tightened sandbox bridged-tool execution to only model-proposed calls, and extended model and computer-use coverage to GPT-6, Grok 4.7, Claude Opus 5.5, Fable, and Mythos, alongside new async log readers and log-directory status inspection.

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

01 **DeepSeek-V4.1-Flash model support** NEW 85

Adds support for DeepSeek-V4.1-Flash via the `deepseek-flash` model name, including image input, with 1M context and 384K max output tokens; retired names `deepseek-v4-flash` and `deepseek-v4-flash-vision-exp` are now recorded as routing to V4.1 Flash.

Run an eval against DeepSeek-V4.1-Flash (with image input) using the new canonical model name.

```
$ inspect eval myTask.py --model deepseek/deepseek-flash
```

— Exact model name, limits, and migration mapping given. 0.3.269

02 **Computer use for additional Claude models** NEW 80

Supports computer use on Claude Opus 5.5, Fable 5/5.1, and Mythos 5/5.1 via Anthropic's `computer_toolset_20260801` toolset; pass `-M computer_toolset=true` to use it on Claude Opus 4.8, Sonnet 5, and Opus 5.

— Named toolset and flag, but no full example. product docs

03 **LiteLLM proxy model provider** NEW 75

Adds a `litellm-proxy` provider for models served through a LiteLLM proxy, reading each model's upstream model for context window, cost, reasoning, and Claude prompt caching.

Route eval traffic through a self-hosted LiteLLM proxy so Inspect inherits context window, cost, and caching metadata automatically.

```
$ inspect eval myTask.py --model litellm-proxy/gpt-4o
```

Point an eval at a self-hosted LiteLLM proxy so Inspect reads the upstream model's context window and cost metadata automatically.

```
$ inspect eval my_task.py --model litellm-proxy/my-hosted-model
```

— Runnable command and named metadata fields included. 0.3.269

---

04 **Reading eval status from a log directory** NEW 75

Reads eval status from a log directory without needing a live eval process, via `inspect ctl --log-dir` (shown against `inspect reads --log-dir` for an S3 path), letting sample status and events be inspected against local or remote log directories.

Inspect a completed eval log directory on S3 for sample status and events without needing a live eval process.

```
$ inspect reads --log-dir s3://my-eval-bucket/runs/2026-09-25/
```

Monitor eval progress across all logs in a directory without attaching to a live run.

```
$ inspect ctl --log-dir ./logs/my-eval-run
```

— Two runnable commands shown against log directories. 0.3.269

---

05 **Bridged sandbox tool call restrictions** BREAKING 70

Sandbox agent bridge host tools are now denied unless the model proposed the call in a bridged generation: host tools exposed via `bridged_tools` execute only for calls the model actually proposed, once per proposal, regardless of whether an approval policy is active. BridgedToolsSpec(require_proposal=False) opts a server out of this restriction.

— Config class name given, no example command. 0.3.269

06 **Quoted CLI --env values preserve literal commas** IMPROVED

65

Quoted YAML and JSON strings in `--env` CLI arguments (e.g. `--env 'NO_PROXY="localhost,127.0.0.1"'`) now preserve commas as literal text instead of being coerced into lists.

— Exact flag and example value given. 0.3.269

07 **sample_abandoned() hook** NEW

60

Adds a sample_abandoned() hook to report a sample cancelled before anything was logged, whether cancelled while queued or before its re-run, so a source waiting on it no longer stalls.

— Named hook and behavior, no usage example. product docs

THINNER COVERAGE BELOW

08 **New model support: GPT-6, Grok 4.7, Claude Opus 5.5** NEW

55

Supports GPT-6 Sol (`gpt-6-sol`) and GPT-6 Luna (`gpt-6-luna`) via the OpenAI provider, Grok 4.7, and Claude Opus 5.5 (`claude-opus-5-5`) via the Anthropic provider, with thinking always enabled and forced tool choice degraded to auto.

— Lists named models but no usage example. product docs

09 **Async eval log readers** NEW

50

Adds `read_eval_log_sample_summaries_async` and three other async log readers to the public `inspect_ai.log` exports.

— Only one of four readers named explicitly. product docs

10 **Faster CLI and import startup** IMPROVED

25

Faster `inspect` CLI startup and `import inspect_ai`.

— No numbers or benchmark given for the speedup. product docs

0.3.269 CODE ↗ LATEST STABLE

Inspect AI 0.3.269 adds a `litellm-proxy` provider, DeepSeek-V4.1-Flash support, and tighter bridged-tool execution controls.

└──▷ GET THIS VERSION

```
$ git clone --branch 0.3.269
https://github.com/UKGovernmentBEIS/inspect_ai.git
# already have the repo? check out this version:
$ git checkout 0.3.269
```

└──▷ **TRY IT**

Route eval traffic through a self-hosted LiteLLM proxy so Inspect inherits context window, cost, and caching metadata automatically.

```
$ inspect eval myTask.py --model litellm-proxy/gpt-4o
```

Run an eval against DeepSeek-V4.1-Flash (with image input) using the new canonical model name.

```
$ inspect eval myTask.py --model deepseek/deepseek-flash
```

Monitor eval progress across all logs in a directory without attaching to a live run.

```
$ inspect ctl --log-dir ./logs/my-eval-run
```

Adds `litellm-proxy` provider for models served through a LiteLLM proxy, reading each model's upstream model for context window, cost, reasoning, and Claude prompt caching.

Adds BridgedToolsSpec(require_proposal=False) opt-out to the bridged-tools contract, where host tools exposed via `bridged_tools` now execute only for calls the model actually proposed, once per proposal, regardless of whether an approval policy is active.

Adds support for DeepSeek-V4.1-Flash via the `deepseek-flash` model name, including image input, with 1M context and 384K max output tokens; retired names `deepseek-v4-flash` and `deepseek-v4-flash-vision-exp` are now recorded as routing to V4.1 Flash.

Quoted YAML and JSON strings in `--env` CLI arguments (e.g. `--env 'NO_PROXY="localhost,127.0.0.1"'`) now preserve commas as literal text instead of being coerced into lists.

Reads eval status from a log directory with `inspect ctl --log-dir`.

WAS THIS USEFUL?  

Comet Opik

Feedback ⓘ 2.2.81 NOTES ↗ RANK

Opik traces, evaluates, and monitors LLM applications and agentic workflows, with datasets, an LLM-as-judge metric library, and production dashboards.

> Opik 2.2.81 adds automated routing of scored traces and spans into annotation queues for human review, plus richer task classification in session analysis.

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

01 **tasks.kind field in session analysis** NEW 45

Adds `tasks.kind` field to `cipx_session_analysis` for richer session-level task classification.

— Names exact field and table but no usage detail 2.2.81

02 **Automated routing to annotation queues** NEW 35

Scored entities (traces and spans) are automatically routed into annotation queues, enabling automated triage for human review.

— Describes behavior but no config or trigger detail 2.2.81

2.2.81 NOTES ↗ LATEST STABLE

Opik 2.2.81 routes scored entities into annotation queues and adds `tasks.kind` to session analysis.

└──▷ **GET THIS VERSION**

```
$ git clone --branch 2.2.81 https://github.com/comet-ml/opik.git
# already have the repo? check out this version:
$ git checkout 2.2.81
```

Routes scored entities into annotation queues, enabling automated triage of scored traces and spans for human review.

Adds `tasks.kind` field to `cipx_session_analysis` for richer session-level task classification.

WAS THIS USEFUL? 👍 👎

Langfuse

Feedback ⓘ v4.46.0 NOTES ↗ RANK

Langfuse provides tracing, evaluation, and monitoring for LLM applications.

> Langfuse v4.46.0 adds skill management, a side-by-side experiment comparison view, formatted JSON display for dataset items, and a larger table page size option.

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

01 **100-row page size for tracing tables** IMPROVED 45

Tracing tables now support a `100-row` page size option, letting users view more traces at once in a single page.

— Names the exact page size but no UI path v4.46.0

02 **Formatted JSON in dataset items** IMPROVED 35

Dataset items now display formatted JSON, making it easier to inspect structured data within items.

— States what changed but no mechanism v4.46.0

03 **Side-by-side comparison in Experiments view** IMPROVED 30

The Experiments view now streamlines side-by-side comparison, though the specific comparison mechanism is not detailed further.

— Brief description without mechanism or UI path v4.46.0

04 **Skill management** NEW 15

Langfuse v4.46.0 adds skill management, though no further detail on scope or mechanism is provided.

— Only named in summary, no detail given v4.46.0

v4.46.0 NOTES ↗ LATEST STABLE

Langfuse v4.46.0 adds skill management, side-by-side experiment comparison, formatted JSON in dataset items, and a 100-row table page size.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v4.46.0
https://github.com/langfuse/langfuse.git
# already have the repo? check out this version:
$ git checkout v4.46.0
```

Adds a 100-row page size option to tracing tables for viewing more traces at once.

Adds formatted JSON display to dataset items for easier inspection of structured data.

Streamlines side-by-side comparison in the Experiments view.

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ changelog · Sep 26, 2026 CHANGELOG ↗

RANK

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

> Braintrust moved Loop (The Debugger) to a persistent, Braintrust-managed runtime and added OpenTelemetry and OpenInference trace ingestion, alongside CLI span plugins, columnar log storage, new SDK instrumentation, attachment-uploader tuning, and provider/config changes.

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

01 **OpenInference tracing support** NEW 75

Maps OpenInference traces to structured messages, tool calls, span types, token metrics, tags, and scores for Braintrust-hosted deployments, supporting OpenInference instrumentors for LangChain, LlamaIndex, and DSPy via OpenTelemetry pipelines (requires data plane v2.16.0+).

— Names instrumentors and version requirement precisely. snapshot-20260926

02 **Attachment uploader tuning via env vars** NEW 75

Adds `BRAINTRUST_ATTACHMENT_UPLOADER_QUEUE_SIZE`, `BRAINTRUST_ATTACHMENT_UPLOADER_REQUEST_TIMEOUT_MILLIS`, `BRAINTRUST_ATTACHMENT_UPLOADER_MAX_RETRIES`, and `BRAINTRUST_ATTACHMENT_UPLOADER_INITIAL_RETRY_DELAY_MILLIS` environment variables, set via `BraintrustConfig`, to tune S3 attachment uploader behavior.

— Exact env var names ready to set immediately. changelog-20260926-867ae438

03 **Trace.getSpans span filtering API** NEW 75

Provides a Trace.getSpans() method to filter captured spans by properties such as error state after a run, for example await Trace.getSpans({ error: true, type: 'tool' }) to surface only failing tool calls for review.

Filter captured spans by error state after a run to surface only failing tool calls for review.

typescript

```
import { Trace } from 'braintrust';

const spans = await Trace.getSpans({ error: true, type: 'tool'
});
console.log(spans);
```

— Runnable code sample with exact method and filters. changelog-20260926-867ae438

04 **CLI span plugins for trace processing** NEW 70

Adds CLI span plugins that run before spans are sent to Braintrust, for example redacting a secret from coding-agent trace spans with `bt trace --plugin ./redact-secrets.js`.

Redact a secret from coding-agent trace spans before they are sent to Braintrust.

```
$ bt trace --plugin ./redact-secrets.js
```

— Runnable command demonstrates exact usage. snapshot-20260926 changelog-20260926-867ae438

05 **OpenTelemetry trace and log ingestion** NEW 65

Supports sending OpenTelemetry traces and logs to Braintrust via an exporter, a Collector, or a Braintrust span processor, enabling Braintrust and OpenTelemetry spans to be linked into a single trace within an application or across services. Maps OpenTelemetry span attributes and events to Braintrust fields, including how log records are stored.

— Describes mechanism and mapping but no exact config shown. product docs

06 **--relabel-overlap flag removed from topics config** BREAKING 65

The `--relabel-overlap` flag is removed from `bt topics config enable` and `bt topics config set`; it had been deprecated and ignored since data plane v2.15.0. See the v0.20.0 migration instructions.

— Exact commands and flag named with migration reference. changelog-20260926-867ae438

07 **Loop debugger runs in managed runtime** IMPROVED 60

Loop (The Debugger) now runs in a Braintrust-managed runtime, enabling longer investigations, persistent threads that span sessions and the full project (logs, experiments, and datasets), and the ability to create and edit project objects such as prompts, scorers, datasets, facets, custom views, dashboards, and automations.

— Describes behavior change but gives no UI path or command. snapshot-20260926 changelog-20260926-867ae438

08 **Columnar log storage for SQL-style queries** NEW 60

Adds columnar log storage, allowing SQL-style queries over logs such as `SELECT DISTINCT user_id, model_name FROM logs WHERE project_id = '<project_id>'` to deduplicate results and count distinct user cohorts.

Deduplicate SQL query results by a unique column combination to count distinct user cohorts in a log query.

```
$ SELECT DISTINCT user_id, model_name FROM logs WHERE project_id
= '<project_id>'
```

— Runnable SQL query demonstrates the capability. snapshot-20260926

THINNER COVERAGE BELOW

09 **Project providers scoped to explicitly served models** BREAKING

55

On Braintrust-hosted deployments, project providers now override organization providers only for models they explicitly serve; organization providers can serve other models using their own credentials. Review organization provider model selections and permissions if you rely on project providers to restrict model access or spending.

— Explains behavior change with review guidance, no exact API. snapshot-20260926 changelog-20260926-867ae438

10 **TypeSafe SDK instrumentation for Python and TypeScript** NEW

50

Adds TypeSafe instrumentation in the Python SDK (v0.41.0+) for synchronous and asynchronous SDK calls, and TypeSafe instrumentation and tracing for experimental_evaluate() calls in the TypeScript SDK (v3.34.0+).

— Names SDK versions but no usage example given. changelog-20260926-867ae438

11 **Models tab in AI provider setup** NEW

50

Adds a new Models tab in AI provider setup to browse available models and copy a Gateway command to test a specific model.

— UI path given, no deeper mechanism described. snapshot-20260926

12 **Node PATH now required for Claude Code and Codex plugins**

50

BREAKING

`node` must be on the PATH of the environment that launches Claude Code, and the Claude Code plugin no longer installs it when missing; likewise `node` must be on the PATH for Codex tracing, as the Codex plugin no longer installs it when missing.

— Clear operational requirement named for two plugins. changelog-20260926-867ae438

13 **Anthropic context_management captured in spans** NEW

45

Captures the `context_management` config and `applied_edits` from Anthropic responses together in span metadata.

— Names fields but gives no usage guidance. changelog-20260926-867ae438

14 **Patterns discovery** NEW 10

Adds a Patterns discovery capability; no further detail on its behavior was published in this window.

— Only named in a summary, no detail available. snapshot-20260926

15 **Jev structured decisions** NEW 10

Adds Jev structured decisions; no further detail on its behavior was published in this window.

— Only named in a summary, no detail available. snapshot-20260926

└──▷ **BREAKING ON UPGRADE**

! The `--relabel-overlap` flag is removed from `bt topics config enable` and `bt topics config set`; it was deprecated and ignored since data plane v2.15.0 — see the v0.20.0 migration instructions.

! On Braintrust-hosted deployments, project providers now override organization providers only for models they explicitly serve; organization providers can serve other models using their own credentials — review organization provider model selections and permissions if you rely on project providers to restrict model access or spending.

! `node` must be on the PATH of the environment that launches Claude Code; the Claude Code plugin no longer installs it when missing.

! `node` must be on the PATH for Codex tracing; the Codex plugin no longer installs it when missing.

! On Braintrust-hosted deployments, project providers now override organization providers only for models they explicitly serve; organization providers can serve other models using their own credentials. If you rely on project providers to restrict model access or spending across all models, review organization provider model selections and permissions.

changelog · Sep 26, 2026 CHANGELOG ↗ LATEST STABLE

Braintrust: Adds `BRAINTRUST_ATTACHMENT_UPLOADER_QUEUE_SIZE`, `BRAINTRUST_ATTACHMENT_UPLOADER_REQUEST_TIMEOUT_MILLIS`, `BRAINTRUST_ATTACHMENT_UPLOADER_MAX_RETRIES`, and `BRAINTRUST_ATTACHMENT_UPLOADER_INITIAL_RETRY_DELAY_MILLIS` environment variables (via `BraintrustConfig`) to tune S3 attachment uploader behavior.

└──▷ **TRY IT**

Redact a secret from coding-agent trace spans before they are sent to Braintrust.

```
$ bt trace --plugin ./redact-secrets.js
```

Filter captured spans by error state after a run to surface only failing tool calls for review.

```typescript
import { Trace } from 'braintrust';

const spans = await Trace.getSpans({ error: true, type: 'tool' });
console.log(spans);
```

Adds TypeSafe instrumentation in the Python SDK (v0.41.0+) for synchronous and asynchronous SDK calls.

Adds TypeSafe instrumentation and tracing for experimental_evaluate() calls in the TypeScript SDK (v3.34.0+).

Upgrades Loop (The Debugger) to run in a Braintrust-managed runtime, enabling longer investigations, persistent threads, cross-workspace access to logs, experiments, and datasets, and the ability to create and edit prompts, scorers, datasets, facets, custom views, dashboards, and automations.

Captures `context_management` config and `applied_edits` from Anthropic responses together in span metadata.

└──▷ **BREAKING ON UPGRADE**

- ! The `--relabel-overlap` flag is removed from `bt topics config enable` and `bt topics config set`; it was deprecated and ignored since data plane v2.15.0 — see the v0.20.0 migration instructions.
- ! On Braintrust-hosted deployments, project providers now override organization providers only for models they explicitly serve; organization providers can serve other models using their own credentials — review organization provider model selections and permissions if you rely on project providers to restrict model access or spending.
- ! `node` must be on the PATH of the environment that launches Claude Code; the Claude Code plugin no longer installs it when missing.
- ! `node` must be on the PATH for Codex tracing; the Codex plugin no longer installs it when missing.

snapshot-20260926 NOTES ↗ STABLE

Braintrust adds OpenInference tracing, CLI span plugins, Patterns discovery, Debugger, Jev structured decisions, and columnar log storage.

└──▷ **TRY IT**

Deduplicate SQL query results by a unique column combination to count distinct user cohorts in a log query.

```
$ SELECT DISTINCT user_id, model_name FROM logs WHERE project_id =
'<project_id>'
```

Maps OpenInference traces to structured messages, tool calls, span types, token metrics, tags, and scores for Braintrust-hosted deployments, supporting OpenInference instrumentors for LangChain, LlamaIndex, and DSPy via OpenTelemetry pipelines (requires data plane v2.16.0+).

Adds a new Models tab in AI provider setup to browse available models and copy a Gateway command to test a specific model.

Moves Loop to a Braintrust-managed server-side runtime so investigations persist across sessions, span the full project (logs, experiments, and datasets), and can create and edit project objects such as prompts.

└──▷ **BREAKING ON UPGRADE**

! On Braintrust-hosted deployments, project providers now override organization providers only for models they explicitly serve; organization providers can serve other models using their own credentials. If you rely on project providers to restrict model access or spending across all models, review organization provider model selections and permissions.

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

NVIDIA OpenShell

Feedback ⓘ v0.1.0 NOTES ↗ RANK

OpenShell runs autonomous AI agents in policy-controlled sandboxes with isolated filesystem, network, and credential access.

> OpenShell v0.1.0 adds new sandbox exec and VM creation controls, server-side sandbox templates, configurable OCSF schema versioning, EC/EdDSA JWKS support, exact MCP revision allowlists in policy, authoritative provider profiles, and expanded middleware HTTP interception, alongside a Helm chart split and compute-driver architecture changes.

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

01 **No-login-shell flag for sandbox exec** NEW 80

The `sandbox exec` command gains a `--no-login-shell` flag that skips shell startup files (`.bashrc` , `.profile` , etc.) when entering a sandbox, useful for clean, reproducible script execution unaffected by a user's shell profile.

Enter a sandbox without running startup files — useful for clean, reproducible script execution that should not be affected by a user's shell profile

```
$ openshell sandbox exec my-sandbox --no-login-shell -- bash -c
'python main.py'
```

— Exact flag, files, and runnable example given. v0.1.0

02 **Authoritative provider profiles via import** IMPROVED 75

Provider profiles are now authoritative: the gateway serves exactly the profiles imported via `openshell provider profile import` .

— Exact command given for provider profile import. v0.1.0

03 **Tar archive source for VM sandboxes** NEW 65

The VM driver adds `--from` source support for rootfs tar archives, enabling VM sandbox creation from a tar image instead of a container image.

Create a VM-backed sandbox from a rootfs tar archive instead of a container image

```
$ openshell sandbox create --from rootfs.tar -- bash
```

— Flag and runnable example given, limited further detail. v0.1.0

THINNER COVERAGE BELOW

04 **Corporate proxy egress for microVM sandboxes** NEW 45

Adds corporate HTTP forward proxy egress support for microVM sandboxes, letting sandbox network traffic route through an organization's HTTP proxy.

— Names microVM and proxy type, no config key given. v0.1.0

05 **Exact MCP revision allowlists in policy** NEW 45

Policy now supports exact MCP revision allowlists, preserving pinned tool versions.

— Names MCP and policy area, no syntax given. v0.1.0

06 **Build, CI and release engineering improvements** IMPROVED 45

Adds a `defaults-without-telemetry` feature alias for builds that omit telemetry, adds Codex Security release qualification to the CI pipeline, and makes the `e2e:kubernetes` target work transparently on OpenShift.

— Names alias, CI check, and test target verbatim. v0.1.0

07 **EC and EdDSA keys in JWKS validation** NEW 40

The server's OIDC JWKS validation now supports EC and EdDSA key types.

— Names algorithms but no endpoint or config given. v0.1.0

08 **Split Helm chart into gateway and workspace** IMPROVED 40

The Helm chart is split into separate gateway and workspace charts.

— Names new chart split but no migration steps. v0.1.0

09 **HTTP response interception and header mutation in middleware** NEW 40

Middleware gains an HTTP response pre-return interface for intercepting and mutating responses before they are returned, and broadens HTTP header mutation authority in middleware.

— Describes new interface, no concrete API name given. v0.1.0

10 **Configurable OCSF schema version** NEW 35

OpenShell adds a configurable OCSF schema version setting for SIEM backward compatibility.

— Names OCSF and SIEM but no config key given. v0.1.0

11 **Compute driver auth and capability delegation** NEW

35

Sandbox authentication is now delegated to compute drivers, enabling driver-owned auth flows, and compute drivers advertise their resource capabilities to the gateway.

— Describes architecture shift without concrete API or config. v0.1.0

12 **Reusable sandbox templates on server** NEW

30

The server now supports sandbox templates, allowing pre-defined sandbox configurations to be reused.

— No config keys or usage path given. v0.1.0

13 **Structured output for list commands** NEW

25

CLI `list` commands now support structured output for machine-readable CLI results.

— No format names or examples given. v0.1.0

v0.1.0 NOTES ↗ LATEST STABLE

OpenShell v0.1.0 adds sandbox templates, structured list output, VM proxy egress, OCSF schema versioning, and EC/EdDSA JWKS support

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.1.0 https://github.com/NVIDIA/OpenShell.git
# already have the repo? check out this version:
$ git checkout v0.1.0
```

└──▷ **TRY IT**

Enter a sandbox without running startup files — useful for clean, reproducible script execution that should not be affected by a user's shell profile

```
$ openshell sandbox exec my-sandbox --no-login-shell -- bash -c
'python main.py'
```

Create a VM-backed sandbox from a rootfs tar archive instead of a container image

```
$ openshell sandbox create --from rootfs.tar -- bash
```

Adds `--no-login-shell` flag to sandbox exec to skip shell startup files (`.bashrc` , `.profile` , etc.) when entering a sandbox

> Adds `--from` source support for rootfs tar archives in the VM driver, enabling VM sandbox creation from a tar image
>
> Adds structured output to `list` commands for machine-readable CLI results
>
> Adds sandbox templates on the server, allowing pre-defined sandbox configurations to be reused
>
> Adds configurable OCSF schema version for SIEM backward compatibility
>
> +12 more

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Frontier Models

Anthropic

Feedback ⓘ changelog · Sep 26, 2026 CHANGELOG ↗ RANK

Anthropic provides Claude AI models and APIs for building applications that generate, analyze, and automate work.

> Anthropic launched Claude Opus 5.5, a long-running agentic coding model, alongside new context-compaction and inline mid-conversation tool support, cache diagnostics, and several Compliance API changes.

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

01 **Claude Opus 5.5 model launch and behavior** NEW 90

`claude-opus-5-5` is a model for long-running agentic coding and knowledge work with a 1M token context window, 128k max output tokens, and always-on adaptive thinking, priced at $4/$20 USD per MTok (cheaper than Claude Opus 5 at $5/$25). Thinking cannot be disabled or explicitly enabled: sending `thinking: {"type": "enabled", ...}` or `thinking: {"type": "disabled"}` returns a 400 error, so the `thinking` field must be omitted and depth controlled via `budget_tokens` or the `effort` parameter instead. `tool_choice` types `any` and `tool` also return a 400 error on this model (use `auto` with strict tool use), and on the Claude API and Google Cloud, computer use requires the `computer_toolset_20260801` toolset since the earlier `computer_20251124` tool now returns a 400 error there (Amazon Bedrock continues to accept `computer_20251124`).

— Names model id, limits, pricing and exact config workarounds `snapshot-20260926` `changelog-20260926-27253774`

02 **Inline mid-conversation tool addition with MCP support** NEW 85

Tools can now be defined inside a mid-conversation system message via a `tool_addition` block (`tool: {"type": "tool_definition", "definition": {...}}`), in beta with the `inline-tools-2026-09-15` header, allowing tool additions, schema changes, or version upgrades without invalidating the prompt cache. MCP toolsets can also be supplied inside `tool_addition` blocks when combining the `inline-tools-2026-09-15` and `mcp-client-2026-09-15` beta headers, with responses recording each server's fetched tool list in an `mcp_tool_listing` block.

Add a new tool mid-conversation without invalidating the prompt cache or rewriting the top-level `tools` array.

```
$ curl https://api.anthropic.com/v1/messages \
  -H 'x-api-key: $ANTHROPIC_API_KEY' \
  -H 'anthropic-version: 2023-06-01' \
  -H 'anthropic-beta: inline-tools-2026-09-15' \
  -H 'content-type: application/json' \
  -d '{
    "model": "claude-opus-5-5",
    "max_tokens": 1024,
    "system": [
      {"type": "tool_addition", "tool": {"type":
"tool_definition", "definition": {"name": "run_query",
"description": "Execute a SQL query", "input_schema": {"type":
"object", "properties": {"query": {"type": "string"}},
"required": ["query"]}}}}
    ],
    "messages": [{"role": "user", "content": "Run a query to
count active users."}]
  }'
```

— Named headers, blocks and a runnable example `snapshot-20260926`

`changelog-20260926-27253774`

03 **Opt-in cache diagnostics on Messages requests** IMPROVED

70

Cache diagnostics can now be requested by sending a `diagnostics` object on a Messages API request, with the beta header no longer required. The API stores a request fingerprint only when the request includes the `diagnostics` object; a request that sends only the beta header stores no fingerprint, so a later turn pointing `previous_message_id` at it reports `previous_message_not_found`.

Opt in to cache diagnostics on a Messages request now that the beta header is no longer required.

```
$ curl https://api.anthropic.com/v1/messages \
  -H 'x-api-key: $ANTHROPIC_API_KEY' \
  -H 'anthropic-version: 2023-06-01' \
  -H 'content-type: application/json' \
  -d '{
    "model": "claude-opus-5-5",
    "max_tokens": 512,
    "diagnostics": {},
    "messages": [{"role": "user", "content": "Hello"}]
  }'
```

— Names the field, error code, and includes a runnable example

changelog-20260926-27253774

04 **Mid-conversation context compaction** NEW 65

A long agentic session can be compacted mid-conversation to stay within context limits without losing earlier context, by sending a `compaction: {}` object on a Messages API request with the `compact-2026-09-04` beta header.

Compact a long agentic session mid-conversation so that subsequent turns stay within context limits without losing earlier context.

```
$ curl https://api.anthropic.com/v1/messages \
  -H 'x-api-key: $ANTHROPIC_API_KEY' \
  -H 'anthropic-version: 2023-06-01' \
  -H 'anthropic-beta: compact-2026-09-04' \
  -H 'content-type: application/json' \
  -d '{
    "model": "claude-opus-5-5",
    "max_tokens": 1024,
    "compaction": {},
    "messages": [{"role": "user", "content": "Summarize what we
have done so far."}]
  }'
```

— Runnable curl example with exact header and field changelog-20260926-27253774

THINNER COVERAGE BELOW

05 **Compliance API local sessions graduate for Microsoft 365 apps**

IMPROVED 50

Compliance API local session endpoints are graduated out of beta for Claude for Microsoft 365 sessions in Excel, PowerPoint, Word, and Outlook, identified by `product_surface` values beginning with `office_agents`.

— Names apps and field prefix but no endpoint path snapshot-20260926

06 **Compliance Activity Feed drops file and document names**

50

BREAKING

The Compliance Access Key Activity Feed's `filename` and `title` fields on file, project document, and artifact activities are now always empty or omitted — including for activities recorded before this change — with lookup instead available via a Compliance Access Key holding the `read:compliance_user_data` scope.

— Names fields and scope but no endpoint given snapshot-20260926

changelog-20260926-27253774

07 **Fast mode research preview for Opus 5.5** NEW 20

Enables Fast mode (research preview) for Claude Opus 5.5 on the Claude API.

— Only a name, no mechanism or usage detail given snapshot-20260926

08 **Frontier LLM refusal billing change** BREAKING 20

Refusals before any output when the category is `frontier_llm` are now billed at the rates of the model that ran the request.

— Single sentence with no further mechanism given changelog-20260926-27253774

└──▷ **BREAKING ON UPGRADE**

- ! On Claude Opus 5.5 (`claude-opus-5-5`), sending `thinking: {"type": "enabled", ...}` returns a 400 error — thinking cannot be disabled or explicitly enabled; omit the `thinking` field entirely and use `budget_tokens` to control depth.
- ! The Compliance Access Key activity object's `filename` and `title` fields on file, project document, and artifact activities are now always empty or omitted — including for activities recorded before this change.
- ! For the diagnostics beta, the API now stores a request fingerprint only when the request includes the `diagnostics` object; a request that sends only the beta header stores no fingerprint, and a later turn pointing `previous_message_id` at it reports `previous_message_not_found` .
- ! Refusals before any output when the category is `frontier_llm` are now billed at the rates of the model that ran the request.
- ! On `claude-opus-5-5` , thinking cannot be disabled: sending `thinking: {"type": "disabled"}` or `thinking: {"type": "enabled", ...}` returns a 400 error; omit the `thinking` field and use the `effort` parameter instead.
- ! On `claude-opus-5-5` , `tool_choice` types `any` and `tool` return a 400 error; use `auto` with strict tool use.
- ! On the Claude API and Google Cloud, computer use on `claude-opus-5-5` requires the `computer_toolset_20260801` toolset; the earlier `computer_20251124` tool returns a 400 error (Amazon Bedrock continues to accept `computer_20251124`).
- ! The Compliance API Activity Feed `filename` and `title` fields are now always empty or omitted on all activities, including those recorded before this change.

changelog · Sep 26, 2026 CHANGELOG ↗ LATEST STABLE

Anthropic: Introduces Claude Opus 5.5 (`claude-opus-5-5`), a model for long-running agentic coding and knowledge work, at $4/$20 USD per MTok (cheaper than Claude Opus 5 at $5/$25).

└──▷ **TRY IT**

Compact a long agentic session mid-conversation so that subsequent turns stay within context limits without losing earlier context.

```
$ curl https://api.anthropic.com/v1/messages \
  -H 'x-api-key: $ANTHROPIC_API_KEY' \
  -H 'anthropic-version: 2023-06-01' \
  -H 'anthropic-beta: compact-2026-09-04' \
  -H 'content-type: application/json' \
  -d '{
    "model": "claude-opus-5-5",
    "max_tokens": 1024,
    "compaction": {},
    "messages": [{"role": "user", "content": "Summarize what we have
done so far."}]
  }'
```

Add a new tool mid-conversation without invalidating the prompt cache or rewriting the top-level `tools` array.

```
$ curl https://api.anthropic.com/v1/messages \
  -H 'x-api-key: $ANTHROPIC_API_KEY' \
  -H 'anthropic-version: 2023-06-01' \
  -H 'anthropic-beta: inline-tools-2026-09-15' \
  -H 'content-type: application/json' \
  -d '{
    "model": "claude-opus-5-5",
    "max_tokens": 1024,
    "system": [
      {"type": "tool_addition", "tool": {"type": "tool_definition",
"definition": {"name": "run_query", "description": "Execute a SQL
query", "input_schema": {"type": "object", "properties": {"query":
{"type": "string"}}, "required": ["query"]}}}}
    ],
    "messages": [{"role": "user", "content": "Run a query to count
active users."}]
  }'
```

Opt in to cache diagnostics on a Messages request now that the beta header is no longer required.

```
$ curl https://api.anthropic.com/v1/messages \
  -H 'x-api-key: $ANTHROPIC_API_KEY' \
  -H 'anthropic-version: 2023-06-01' \
  -H 'content-type: application/json' \
  -d '{
    "model": "claude-opus-5-5",
    "max_tokens": 512,
    "diagnostics": {},
    "messages": [{"role": "user", "content": "Hello"}]
  }'
```

└──▷ **BREAKING ON UPGRADE**

- ! On Claude Opus 5.5 (`claude-opus-5-5`), sending `thinking: {"type": "enabled", ...}` returns a 400 error — thinking cannot be disabled or explicitly enabled; omit the `thinking` field entirely and use `budget_tokens` to control depth.
- ! The Compliance Access Key activity object's `filename` and `title` fields on file, project document, and artifact activities are now always empty or omitted — including for activities recorded before this change.
- ! For the diagnostics beta, the API now stores a request fingerprint only when the request includes the `diagnostics` object; a request that sends only the beta header stores no fingerprint, and a later turn pointing `previous_message_id` at it reports `previous_message_not_found` .
- ! Refusals before any output when the category is `frontier_llm` are now billed at the rates of the model that ran the request.

`snapshot-20260926` **NOTES ↗** **STABLE**

Anthropic: Launches `claude-opus-5-5` , a model for long-running agentic coding and knowledge work with a 1M token context window, 128k max output tokens, and always-on adaptive thinking at $4/$20 per MTok.

Enables Fast mode (research preview) for Claude Opus 5.5 on the Claude API.

Supports defining tools inside a mid-conversation system message via a `tool_addition` block (with `tool: {"type": "tool_definition", "definition": {...}}`), in beta on the Claude API with the `inline-tools-2026-09-15` beta header — allowing tool additions, schema changes, or version upgrades without invalidating the prompt cache.

Supports MCP toolsets inside `tool_addition` blocks when using both `inline-tools-2026-09-15` and `mcp-client-2026-09-15` beta headers; responses record each server's fetched tool list in an `mcp_tool_listing` block.

Graduates Compliance API local session endpoints out of beta for Claude for Microsoft 365 sessions in Excel, PowerPoint, Word, and Outlook (`product_surface` values beginning with `office_agents`).

Removes file names, project document names, and artifact titles from the Compliance API Activity Feed; the `filename` and `title` fields are now always empty or omitted, with lookup available via a Compliance Access Key with the `read:compliance_user_data` scope.

└──▷ **BREAKING ON UPGRADE**

- ! On `claude-opus-5-5` , thinking cannot be disabled: sending `thinking: {"type": "disabled"}` or `thinking: {"type": "enabled", ...}` returns a 400 error; omit the `thinking` field and use the `effort` parameter instead.
- ! On `claude-opus-5-5` , `tool_choice` types `any` and `tool` return a 400 error; use `auto` with strict tool use.
- ! On the Claude API and Google Cloud, computer use on `claude-opus-5-5` requires the `computer_toolset_20260801` toolset; the earlier `computer_20251124` tool returns a 400 error (Amazon Bedrock continues to accept `computer_20251124`).
- ! The Compliance API Activity Feed `filename` and `title` fields are now always empty or omitted on all activities, including those recorded before this change.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

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

▷ Subscribe | Feedback on this issue