THE AI TOOLCHAIN

NO. 051

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED OCTOBER 8, 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 72 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 9 tools

▾ // THE BRIEF HIDE

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

Three unrelated things stand out. Claudoscope puts a time axis and Replay on agent sessions, so a bad run can be stepped through instead of reconstructed from logs. Strands Agents makes its bidirectional

vim ? help

streaming agent GA and adds agent.cancel(). And gridctl imports existing MCP server configs with secrets encrypted on the way in.

> OPERATE
>
> **Step through a bad agent session on a timeline instead of scrolling the transcript**
>
> Claudoscope 1.4's Trace tab lays a session on a time axis with Replay, a minimap and persistent bookmarks, so the moment a run went wrong is a position you can jump back to rather than a point you hunt for by scrolling. Langfuse v4.55.0's batch score ingestion lets you push evaluation scores for many traces in one call instead of one request per score, which is what makes backfilling judgments over a week of production traffic practical.
>
> Claudoscope · Langfuse

> BUILD
>
> **Run a long-lived voice or streaming agent you can interrupt mid-turn**
>
> Strands Agents promotes BidiAgent to GA with agent.cancel() and Agent.shutdown(), so a user talking over the model or a hung tool call can be stopped cleanly rather than waiting out the turn; a vended subagent tool and TypeScript tool ports mean the same tools work across both SDKs. Hermes v0.21.6 adds live speech-to-text streaming with dedicated voice-turn routing, so transcription and the model response are not serialized behind a finished utterance.
>
> Strands Agents · Hermes

> GOVERN
>
> **Onboard MCP servers and plugins without hand-copying secrets or trusting the code**
>
> gridctl imports MCP server configs that already exist on a machine and encrypts their secrets automatically, and its skill import/sync keeps lockfiles that show drift across AI clients, so you can see which workstation has quietly diverged. Hermes adds a per-profile plugin sandbox, which confines third-party plugin code to the profile that loaded it instead of giving it the whole session.
>
> gridctl · Hermes

BUILD

◆ AI Coding Agents

yetone magpie

Feedback ⓘ v0.1.1122 CODE ↗ RANK

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

> magpie's latest releases add Qwen Code and Alma as managed agents, a Codex-shaped `/v1/codex/models` endpoint for remote discovery, per-provider requests-per-minute limiting, real-time Codex usage-cap enforcement, and richer account/usage tracking across its Routing and Settings UI.

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

01 **Qwen Code agent integration** NEW 90

magpie adds Qwen Code (`qwen` , QwenLM/qwen-code) as a managed agent: it appends one `modelProviders.openai` entry per catalog model into `~/.qwen/settings.json` , sets `settings.model` and `security.auth.selectedType` to `openai` , injects the gateway key via `MAGPIE_QWEN_API_KEY` in the settings `env` object, uses display names in `magpie/<provider>/<model>` format and wire ids as the gateway expects, and restores the user's own configuration on disconnect.

Override the Qwen Code gateway key magpie injects by exporting the same variable name before launching qwen, so your shell value wins over the settings file.

```
$ export MAGPIE_QWEN_API_KEY=<your-key>
qwen
```

— Exact config paths and env var, plus a runnable override example v0.1.1120

02 **Codex-shaped model catalog endpoint** NEW 88

Adds `GET /v1/codex/models` , returning magpie's model catalog in Codex's own shape so a remote Codex instance can point its `model_catalog_url` at magpie's gateway; `/v1/models` remains OpenAI-compatible and unchanged.

Point a remote Codex instance at magpie's gateway so it discovers all of magpie's models in Codex's native catalog format.

```
$ curl -H 'Authorization: Bearer <gateway-key>' https://<magpie-host>/v1/codex/models
```

— Full endpoint with runnable curl example v0.1.1118

03 **Per-profile omp agent rows** UPDATED INTEGRATION NEW 78

Profiles under `~/.omp/profiles/*` (or `~/$PI_CONFIG_DIR/profiles`) each become an independent agent row — `omp#<name>` on Windows, `omp#<name>@wsl:<distro>` in a WSL distro — each with its own `config.yml` and `models.yml`.

— Exact paths and naming scheme but no CLI invocation v0.1.1118

04 **Alma agent integration and MCP wiring** NEW 74

Integrates `alma-server` on Linux as the Alma agent so its requests are identified as Alma rather than ai-sdk, with the model magpie draws using Alma's image generation model when Alma is set to Auto. Reads MCP server config from `~/.config/alma/mcp.json` and supplies it to Alma as its own MCP settings.

— Exact config path named but no CLI command shown v0.1.1119 v0.1.1118

05 **Account usage tracking and quota cards** IMPROVED 73

Remote magpie account cards (`CachedCards`) are enriched with live readings pulled from account switching, Codex warm-up (`LoginUsage`), and routing (Allowances, `KeyAllowance`), all persisted across restarts in `quotas.json`. The subscription card window summary now shows 'Whole ≈ N tokens · ≈ $X', totalling all tokens routed through the account since the window began.

— Names exact data fields and storage file but no UI path v0.1.1120

06 **Per-provider requests-per-minute limit** NEW 70

Adds a Requests per minute limit to providers so each key or account sends the vendor at most that many requests in any rolling 60-second window; requests beyond the cap wait or receive a 429 response with a Retry-After header.

— Clear mechanism and response code but no exact setting location v0.1.1119

07 **Real-time Codex usage-cap enforcement** NEW 63

Enforces Codex account usage caps in real time: each ChatGPT reply's `codex.rate_limits` event and `x-codex-*` headers update the account's reading, holding a capped account from the next turn onward.

— Mechanism named but no way to configure the cap v0.1.1118

THINNER COVERAGE BELOW

08 **Daily stats agent/provider/model usage reporting** NEW 52

Daily stats events now report which agents, providers, and models magpie is used with, using magpie's own IDs (custom providers and their models are recorded as custom),

along with usage counts.

— Names data captured but no access point given v0.1.1118

09 **Missing model detection in Settings** IMPROVED 50

Surfaces missing image-recognition or image-generation models in Settings › Models: a model picked there that magpie can no longer find is marked missing rather than silently replaced.

— Names UI path but no further mechanism v0.1.1118

10 **Tool-only Responses model routing** IMPROVED 45

Routes a model that a relay serves with tools on the Responses API alone to the provider's Responses URL rather than the default endpoint.

— Describes behaviour but no config surface or example v0.1.1120

11 **Routing view session labels and image-description reasoning** IMPROVED 45

The Routing By session view now labels each Claude Code (and other agent) session with its own name instead of its UUID. Routing also now explains why an image was described for a model, noting whether the model is counted as text-only or whether the provider's list was consulted.

— Describes UI change without exact navigation steps v0.1.1118

NEW INTEGRATION Sponsored partner provider feed

v0.1.1122 CODE ↗ LATEST STABLE

Routing UI folds 3+ keys for one model into a collapsible row; `MAGPIE_PARTNERS=off` disables the partner feed; Codex Ultrafast falls back to Fast on unsupported plans.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.1122 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.1122
```

Adds `MAGPIE_PARTNERS` environment variable to override the partner list source URL or set it to `off` to disable the partner feed entirely.

Routing view folds three or more of a provider's keys for one model into a single collapsible row showing the count and state, with open/closed state persisted via `magpie.routingKeysOpen` in `localStorage`.

> Gateway: Codex Ultrafast routing now targets a ChatGPT account whose model list offers it, and falls back to Fast for accounts on plans that do not support it.
>
> Usage view shows a DeepSeek balance in both CNY and USD simultaneously, ordered by the currency with funds remaining first.

WAS THIS USEFUL? 👍 👎

Claudoscope

Feedback ⓘ v1.4.0 NOTES ↗ RANK

Claudoscope is a native macOS menu-bar app that tracks Claude Code sessions, estimates costs, scans secrets, and surfaces analytics.

> Claudoscope 1.4 introduces a time-axis Trace tab with Replay and a shareable summary card, adds a session minimap and persistent bookmarks, brings in-app session resume, and updates pricing and menu-bar indicators.

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

01 **Bookmarks for session turns** NEW 85

Right-clicking any user or assistant turn bookmarks it with an optional note; bookmarks persist in the local SQLite cache and surface as a sidebar badge, a turn accent in Chat, a minimap mark, a cross-session Bookmarks section in the sidebar, and via the command palette (Cmd+K).

— Storage mechanism, shortcut, and five surface points named v1.4.0

02 **Resume session in terminal** NEW 85

A Resume button in the session header runs `claude --resume <id>` and opens a new terminal window (Ghostty, iTerm2, or Terminal.app) in the session's working directory, so a prior session can be reopened without hunting for its ID.

Jump back into a previous Claude Code session from inside Claudoscope without hunting for the session ID yourself.

```
$ claude --resume <id>
```

— Exact command, button, and supported terminals given v1.4.0

03 **Menu-bar live session count badge** NEW 85

A 'Show live session count in the menu bar' toggle in Settings > Appearance (off by default) renders a badge next to the icon — blue normally, amber when waiting on you, red when a wait escalates, and a plain bold digit in monochrome mode.

— Exact settings path and color states specified v1.4.0

04 **Replay playback in Trace tab** NEW 80

Replay in the Trace tab sweeps a playhead across the session at 60×, 300×, 1200×, or fitted to twenty seconds, showing current turn, context fill, and spend in the footer; Space toggles playback and Escape stops it.

— Exact speeds and keyboard controls given, no command v1.4.0

05 **Trace share card export** NEW 75

A Share card exports the trace as a 1200×630 PNG (save or copy) containing project name, duration, turns, tool calls, agents, compactions, cost, and model family — deliberately omitting transcript text or agent descriptions.

— Exact format and fields named but only a UI action v1.4.0

06 **Haiku 5.5 pricing model** NEW 75

Adds `claude-haiku-5-5` pricing at $0.10/$0.50 per million tokens with $0.01 cache reads and a prompt-length tier: requests over 100K tokens (input plus cached prefix) bill at $0.50/$2.50 for that message, evaluated per message from actual request size.

— Precise rates and tier logic but nothing to configure v1.4.0

07 **Time-axis Trace tab for sessions** NEW 70

A new Trace tab lays a full session out on a time axis — prompts, tool calls by category, subagent forks, context fill, compactions, and cumulative cost — with idle stretches over five minutes collapsing to a labeled break.

— Names tracked signals and idle threshold but only a UI path v1.4.0

08 **Session minimap strip** NEW 65

A minimap strip above the chat shows user/assistant turns, tool calls colored by category, tool errors, blocked actions, compactions, subagent spawns, and bookmarks, with click-to-scroll navigation.

— Lists tracked elements, only UI interaction described v1.4.0

THINNER COVERAGE BELOW

09 **One-time reparse on upgrade** BREAKING 35

First launch after upgrading triggers a one-time full reparse of all session files (parser version 14).

— States the trigger but no migration guidance or duration v1.4.0

└──▷ **BREAKING ON UPGRADE**

! First launch after upgrading triggers a one-time full reparse of all session files (parser version 14).

WAS THIS USEFUL?  

AGENT

◆ MCP Servers & Tooling

gram

Feedback ⓘ server@2.20.0 NOTES ↗ RANK

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

> Gram rebranded its environment variables and headers from Gram-prefixed to Speakeasy AI naming (with a breaking API-key change), and overhauled access controls in the Role editor and Access Hub guided setup.

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

01 Speakeasy AI environment variable rebrand

UPDATED INTEGRATION

95

BREAKING

Generated plugins for Cursor, OpenCode, Codex, and Claude Code now read the API key exclusively from `SPEAKEASY_AI_API_KEY`, with no fallback to the old `GRAM_*` variables, so users must set `SPEAKEASY_AI_API_KEY` after updating plugins or authentication will fail. The credential cache now lives at `$XDG_CONFIG_HOME/speakeasy-ai/hooks-auth.env`, while the previous cache at `$XDG_CONFIG_HOME/gram/hooks-auth.env` continues to work. LiteLLM ingest now reads `speakeasy-ai-session-id`, `speakeasy-ai-agent-provider`, and `speakeasy-ai-turn-id` headers before falling back to the old `x-gram-*` names.

Migrate hooks configuration to the new Speakeasy AI environment variable names so the plugin authenticates correctly after the rename.

```
$ export SPEAKEASY_AI_HOOKS_API_KEY=sk-your-key
export SPEAKEASY_AI_HOOKS_HOME=~/.config/speakeasy-ai/hooks
export SPEAKEASY_AI_API_KEY=sk-your-key
```

— Names exact env vars, headers, and cache paths with migration command.

dashboard@0.134.0

02 Role editor MCP access tab overhaul

IMPROVED

65

The role editor's MCP access tab is now a server picker with fuzzy search, collapsible project groups, per-server tool-level access controls (all tools, a hand-picked list, or annotation-filtered), and a forbidden-servers section.

— Names each new control but no config keys or API. dashboard@0.134.0

THINNER COVERAGE BELOW

03 Access Hub guided setup shows connection details

IMPROVED

55

Access Hub's guided setup now displays the token endpoint, issuer, and MCP host that an external platform is pointed at, with a dropdown listing every user session issuer by slug.

– Describes new UI fields but no exact navigation path. dashboard@0.134.0

04 **environmentSlug replaces gramEnvironment in Elements**

DEPRECATED

45

Elements now supports an `environmentSlug` option as the canonical replacement for `gramEnvironment` when selecting which Gram environment a server points at.

– Names the two option keys but no migration mechanics given. dashboard@0.134.0

05 **Scoped issuer picker for scope overrides** IMPROVED

40

The issuer scope override input is now a picker limited to the provider's supported scopes, replacing the previous free-text field.

– States the change but no UI path or scope list. admin@0.9.0

NEW INTEGRATION Claude Tag Access Hub catalog platform

UPDATED INTEGRATION Okta Integration Network client secret authentication

└──▷ BREAKING ON UPGRADE

! Generated plugins for Cursor, OpenCode, Codex, and Claude Code read the API key exclusively from `SPEAKEASY_AI_API_KEY` with no fallback to `GRAM_*` variables — set `SPEAKEASY_AI_API_KEY` after updating plugins or they will fail to authenticate.

server@2.20.0 NOTES ↗ LATEST STABLE

New API methods for catalog platforms, token endpoints, MCP scope pins, and hook status; canonical Speakeasy AI request headers and env vars replace Gram-* names.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch server@2.20.0 https://github.com/speakeasy-api/gram.git
# already have the repo? check out this version:
$ git checkout server@2.20.0
```

Adds `workloadIdentities.listPlatforms` API method (requires `workload:read`) to serve YAML-defined catalog platform definitions and their guided setup steps.

Adds `workloadIdentities.listTokenEndpoints` API method (requires `workload:read`) to expose token endpoint, issuer, and MCP host for each user session issuer via its `/oauth/usi/{id}` authorization server metadata.

Adds `remoteMcp.getServerScopes` and `remoteMcp.setServerScopePin` API endpoints to read and set a remote-backed MCP server's protected resource scope pin, shared across all servers in the project with the same upstream URL.

Adds `hooks.getStatus` project-read API endpoint to report whether hook telemetry is configured — via an active hooks-scoped API key or an enabled Anthropic inference hooks integration.

Renames canonical request headers to `Speakeasy-AI-Key`, `Speakeasy-AI-Project`, `Speakeasy-AI-Session`, `Speakeasy-AI-Chat-Session`, `Speakeasy-AI-Environment`, `Speakeasy-AI-Mode`, `Speakeasy-AI-User-Email`, `Speakeasy-AI-Device-Serial`, `-Hostname`, and `-Device-Environment`; each replaces its `Gram-*` counterpart and wins when both are sent.

+5 more

└─▷ **BREAKING ON UPGRADE**

- ! Generated plugins (Cursor, OpenCode, Codex, Claude Code) read the API key from `SPEAKEASY_AI_API_KEY` only — there is no fallback to `GRAM_API_KEY` or similar — so `SPEAKEASY_AI_API_KEY` must be set before updating plugins.
- ! The `environmentSlug` option replaces `gramEnvironment` in Elements; `gramEnvironment` remains as a fallback only when `environmentSlug` is not set.

WAS THIS USEFUL? 👍 👎

◆ AI Agent Frameworks

AWS Strands Agents

Feedback ⓘ harness-cli/v0.2.0 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 graduated its bidirectional streaming API (BidiAgent) to general availability with a renamed `auto_restart` config and several breaking removals, while also adding an `aux_model` parameter, agent.cancel()/Agent.shutdown() lifecycle controls, a vended subagent tool for multi-agent pipelines, TypeScript tool ports, CLI additions, and an MCP client upgrade across the Python and TypeScript SDKs.

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

01 Bidirectional streaming (BidiAgent) graduated to GA

BREAKING

90

The `BidiAgent` bidirectional streaming API moves from preview to general availability, adding agent-level storage, cancellation from custom tools, grouped tool-call preservation, text and reasoning stream events, console I/O display of tool calls, normalized token usage events, support for typing and speech sharing a single console, and `BidiAgent` support in `SnapshotSessionManager` for session persistence across streaming sessions. The GA also renames the `auto_reconnect` setting to `auto_restart`, removes session manager configuration from `BidiAgent`, and removes directory tool watching from `BidiAgent` — existing code referencing `auto_reconnect` will break.

— Names the renamed config key and removed features but gives no code example.

harness-cli/v0.2.0 harness-typescript/v0.2.0 typescript/v1.20.0

02 `aux_model` parameter for secondary model calls

NEW

80

Agents gain an `aux_model` parameter enabling a secondary, auxiliary model path alongside the primary agent model — e.g. a cheaper or faster model for summarisation, routing, or classification while the primary model handles main reasoning.

Use a cheaper or faster model for auxiliary calls (e.g. summarisation, routing) while keeping a powerful primary model for main reasoning.

```python
from strands import Agent

agent = Agent(
    model="us.anthropic.claude-opus-4-5",
    aux_model="us.anthropic.claude-haiku-4-5"
)
agent("Summarise and then deeply analyse this codebase")
```

Use a lighter auxiliary model for cheap classification calls while the primary model handles reasoning.

```typescript
import { Agent } from '@strands-agents/sdk'

const agent = new Agent({
  aux_model: 'amazon.nova-micro-v1:0'
})

await agent.invoke('Summarize and classify this support ticket')
```

— Named parameter with runnable Python and TypeScript examples. `harness-cli/v0.2.0`

03 **`agent.cancel()` to stop the event loop** NEW 75

Adds agent.cancel() as the mechanism to stop the agent event loop, letting a long-running agent be interrupted from an external signal handler or timeout without killing the process.

Cancel a long-running agent loop from an external signal handler or timeout without killing the process.

```python
import asyncio
from strands import Agent

agent = Agent()

async def main():
    task = asyncio.create_task(agent.invoke_async("Crawl and
index the entire site"))
    await asyncio.sleep(30)   # hard timeout
    agent.cancel()            # stop the event loop gracefully
    await task

asyncio.run(main())
```

Stop a running TypeScript agent cleanly — call agent.cancel() to halt the event loop mid-turn, then Agent.shutdown() to release all scoped resources when done.

```typescript
import { Agent } from '@strands-agents/sdk'

const agent = new Agent()

// Halt the event loop during a long-running turn
agent.cancel()

// Release resources when the agent scope is finished
await agent.shutdown()
```

Cancel a long-running agent from an external signal or timeout without killing the process.

```typescript
import { Agent } from '@strands-agents/sdk'

const agent = new Agent()

const task = agent.invoke('Analyze the entire codebase for
security issues')

// Cancel after 30 seconds
setTimeout(() => agent.cancel(), 30_000)

await task
```

— Exact method name with runnable cancellation examples in two languages.

harness-cli/v0.2.0

04 **`Agent.shutdown()` for scope-based cleanup** NEW 70

Adds Agent.shutdown() for scope-based cleanup of agent resources, ensuring connections, sessions, and background tasks are released deterministically when a scope exits.

Ensure all agent-held resources (connections, sessions, background tasks) are released deterministically when a scope exits.

python

```python
from strands import Agent

agent = Agent()
try:
    agent("Run the nightly report")
finally:
    await agent.shutdown()
```

Stop a running TypeScript agent cleanly — call agent.cancel() to halt the event loop mid-turn, then Agent.shutdown() to release all scoped resources when done.

typescript

```typescript
import { Agent } from '@strands-agents/sdk'

const agent = new Agent()

// Halt the event loop during a long-running turn
agent.cancel()

// Release resources when the agent scope is finished
await agent.shutdown()
```

— Exact method with runnable cleanup example. harness-python/v0.2.0

05 **`InterruptError` exported in TypeScript SDK** NEW 70

Exports `InterruptError` from the TypeScript SDK so human-in-the-loop interruptions and user-requested stops can be caught and handled explicitly instead of crashing the outer process.

Catch a user-requested stop in TypeScript using the newly exported `InterruptError` so your harness can clean up or surface a partial result.

```typescript
import { createHarness } from '@strands-agents/harness'
import { InterruptError } from '@strands-agents/sdk'

const agent = await createHarness()
try {
  await agent.invoke('Run a full dependency-audit across all
packages')
} catch (err) {
  if (err instanceof InterruptError) {
    console.log('Agent was interrupted — partial results saved.')
  } else {
    throw err
  }
}
```

Catch `InterruptError` in TypeScript to handle human-in-the-loop interruptions without crashing the outer process.

```typescript
import { Agent, InterruptError } from '@strands-agents/sdk'

const agent = new Agent()
try {
  await agent.invoke('Deploy the staging environment')
} catch (err) {
  if (err instanceof InterruptError) {
    console.log('Agent was interrupted — awaiting human approval
before resuming')
  } else {
    throw err
  }
}
```

Catch and handle agent interrupts with type safety using the newly exported InterruptError.

```typescript
import { Agent, InterruptError } from '@strands-agents/sdk'

const agent = new Agent()

try {
  await agent.invoke('Draft and send a production deployment')
} catch (err) {
  if (err instanceof InterruptError) {
    console.log('Agent was interrupted — awaiting human approval
before continuing')
  }
}
```

— Named export with three runnable catch examples. harness-cli/v0.2.0

06 **CLI `strands update`, `/tools`, and setup flow refactor**

IMPROVED 65

Adds a `strands update` subcommand and a `/tools` command to the CLI, refines the `compact`, `skills`, `rename`, and `setup` flows, and replaces the setup Q&A with Quickstart, Customize, and Export modes.

— Names every new subcommand and CLI mode but no example given. harness-typescript/v0.2.0 harness-python/v0.2.0

THINNER COVERAGE BELOW

07 **`@strands-agents/sdk` becomes a peer dependency** BREAKING 55

`@strands-agents/sdk` is now a `peerDependency` of `harness-ts` rather than a direct dependency, requiring it to be installed separately.

— Exact package name and dependency change with clear install implication. harness-cli/v0.2.0

08 **MCP TypeScript client upgraded to 2.0** BREAKING 50

The MCP TypeScript client is upgraded to `@modelcontextprotocol/client` 2.0; existing MCP integrations depending on the 1.x client interface may break.

— Names exact package and version but no migration steps. typescript/v1.20.0

09 **Vended subagent tool for multi-agent pipelines** NEW 45

Adds a vended subagent tool, allowing an agent to be packaged and invoked as a tool by another agent, so agents can spin up managed subagents as tools within multi-agent pipelines.

— Describes capability but gives no example or config. `python/v1.59.0` `harness-cli/v0.2.0`

---

10 **`mcp_router` and `a2a_client` vended tools ported to TypeScript**

NEW

45

Ports the `mcp_router` vended tool and the `a2a_client` vended tool to TypeScript, bringing parity with existing Python vended tool support.

— Names both ported tools but gives no usage detail. `harness-cli/v0.2.0`

---

11 **Strands coding-agent skill for framework scaffolding** NEW

40

Adds a Strands coding-agent skill covering framework selection, scaffolding, tools, MCP integration, and LangGraph migrations.

— Names scope areas but no commands or config shown. `harness-cli/v0.2.0`

---

12 **Tool client accessor on MCP integration** NEW

23

Exposes a tool client accessor on the MCP integration.

— One-line mention with no mechanism or usage shown. `harness-cli/v0.2.0`

---

13 **Agent axis policies migrated in multi-agent subsystem**

IMPROVED

23

Migrates agent axis policies to the SDKs in the multi-agent subsystem.

— Bare mention with no explanation of scope. `harness-cli/v0.2.0`

└──▷ BREAKING ON UPGRADE

! The `auto_reconnect` setting on `BidiAgent` is renamed to `auto_restart`.

! Session manager configuration is removed from `BidiAgent`.

! Directory tool watching is removed from `BidiAgent`.

! `@strands-agents/sdk` is now a `peerDependency` of `harness-ts` rather than a direct dependency, requiring it to be installed separately.

! The `auto_reconnect` config key is renamed to `auto_restart` in the bidirectional streaming API — existing code referencing `auto_reconnect` will break.

! The `auto_reconnect` option on `BidiAgent` is renamed to `auto_restart`.

! The MCP TypeScript client is upgraded to `@modelcontextprotocol/client` 2.0 (marked `!` in the commit); existing MCP integrations depending on the 1.x client interface may break.

WAS THIS USEFUL?  

Nous Research Hermes

Feedback ⓘ v0.21.6 NOTES ↗

CODE ↗ RANK

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

> Hermes v0.21.6 adds live speech-to-text streaming and dedicated voice-turn routing, a per-profile plugin sandbox, a new `hermes status` command, and an expanded model catalog with local model support, alongside numerous smaller UI, safety, and reliability improvements.

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

01 **Per-profile plugin sandbox isolation** NEW 80

The `plugins.isolation: host` config key runs third-party plugins in a per-profile sandbox isolated from the main agent process, paired with a unified Settings ▸ Plugins page for managing their settings.

Sandbox untrusted third-party plugins in a per-profile host so they cannot reach your main agent process.

yaml

```yaml
plugins:
  isolation: host
```

— Config key, example, and UI location all given v0.21.6

02 **Voice chat routing to a dedicated model** NEW 70

The `auxiliary.voice_chat` config key routes spoken voice turns to a dedicated model, with reasoning off by default, keeping the main text model unaffected.

Route all spoken voice turns to a separate model (with reasoning off by default) to keep your text model unaffected.

yaml

```yaml
auxiliary:
  voice_chat: true
```

— Config key with runnable example snippet and clear mechanism v0.21.6

03 **Summary-first `hermes status` command** NEW 70

Adds a summary-first `hermes status` command with `--short` and `--full` flags for checking agent health and scheduled-job status at a glance.

Check overall agent health and scheduled-job status at a glance without the full verbose output.

```
$ hermes status --short
```

— Runnable command with flags and example shown v0.21.6

04 **Expanded model catalog and local model support** NEW 70

Adds GPT-6.1 Sol and Claude Sonnet 5.5 / Haiku 5.5 to the model catalogs, and adds local model support to `hermes model`, including llama.cpp with CUDA on Linux x64 and arm64.

— Named models and command but no setup steps given v0.21.6

05 **Cron store and tick-lock health warnings** NEW 70

Adds `doctor` and home-channel warnings when the cron store goes unwritable, and cron warnings for unwritable tick locks, surfaced in `hermes status` and `doctor`.

— Named surfaces given but no remediation steps v0.21.6

THINNER COVERAGE BELOW

06 **Approval prompts for credential uploads and hidden Unicode** NEW 50

Adds approval prompts for credential uploads and for invisible Unicode detected in terminal commands.

— Triggers named but no config or command to configure v0.21.6

07 **Discord bot setup flow** NEW 50

Adds a Discord setup flow that validates the bot token and prints the invite link.

— Mechanism described but no exact command or path v0.21.6

08 **Diff scope selector in review pane** NEW 45

Adds a review-pane diff scope selector with Uncommitted, Branch, and Last Turn options.

— UI option named with no further mechanism v0.21.6

09 **Fish shell environment activation** NEW 40

Adds `activate.fish` shell support for the Hermes environment.

— Named file but no further usage detail v0.21.6

10 **Pluggable layered language packs** NEW 40

Adds pluggable, layered language packs across core, Desktop, and TUI.

— Surfaces named but no config or command to use them v0.21.6

11 **Favorite and pinned models in Desktop picker** IMPROVED 35

Adds favorite and pinned models to the top of the Desktop model picker.

— Minor UI change with little further detail v0.21.6

12 **Usage chip for subscription consumption** NEW 35

Adds a usage chip showing subscription consumption before a plan limit is reached.

— UI indicator described without thresholds or location v0.21.6

13 **Unified Kanban workflow definition** NEW 20

Adds a unified Kanban workflow definition.

— Bare mention with no operational detail v0.21.6

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Agentic Infrastructure

Unsloth AI Unsloth

Feedback ⓘ v0.1.905-beta NOTES ↗

RANK

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

> Unsloth's v0.1.905-beta release introduces a full decision-model training and serving pipeline via `FastDecisionModel.predict`, adds native ComfyUI model support, and ships OS-level sandboxing alongside faster QLoRA training for large MoE models.

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

01 **FastDecisionModel.predict for decision model training** NEW 95

Adds `FastDecisionModel.predict` for training any text or vision LLM as a Jev-style decision model using QLoRA, with decision accuracy rising from 30% to 80%.

– Names exact API call and reports concrete accuracy gain v0.1.905-beta

02 **OS-level sandboxing for model runs** NEW 75

Sandboxes model-run code with Bwrap on Linux, Seatbelt on Mac, and MXC on Windows, with sandbox status and protection levels visible in Settings.

– Names each OS sandbox mechanism and Settings location v0.1.905-beta

03 **Native ComfyUI model support** NEW 70

Runs supported ComfyUI image and video models directly from Hugging Face, with automatic recognition of ComfyUI checkpoints and local model folders, supports ComfyUI text encoders and VAEs natively, and runs Krea-2, HunyuanImage-2.1 and Wan2.2 expert pairs plus ComfyUI NVFP4 and MXFP8 models.

– Names several supported models and formats but no run command v0.1.905-beta

04 **Faster QLoRA training for large MoE models** IMPROVED 70

Trains Qwen3.5-35B-A3B up to 4.1x faster and Qwen3-30B-A3B up to 3.3x faster with QLoRA on A100 and RTX PRO 6000.

– Concrete speed numbers but no usage steps given v0.1.905-beta

05 **Decision model export, serving and testing** NEW 65

Supports exporting Clef models with Qwen3.5 backbones and Laya models to GGUF format for decision models, serves decision models through llama.cpp including vision-capable ones, saves and resumes smaller adapter and decision-head checkpoints for decision model training, and enables testing trained decision models directly from the Decision API settings.

— Groups several thin decision-model additions, names formats and settings path

v0.1.905-beta

06 **INT8 ConvRot for Qwen-Image-2.1** IMPROVED 60

Enables INT8 ConvRot for Qwen-Image-2.1 by default, preserving more full-precision image detail and accelerating generation on supported NVIDIA GPUs.

— Describes default behavior with no config flag given v0.1.905-beta

07 **Full PEFT init_lora_weights support** IMPROVED 60

Supports every PEFT `init_lora_weights` option, including fast PiSSA and MiCA init.

— Names exact PEFT flag and the init options it covers v0.1.905-beta

THINNER COVERAGE BELOW

08 **Desktop Browser interaction enhancements** NEW 55

Enables asking about open pages in the Desktop Browser, converts `[1]` `-style` reply citations into clickable links, allows confirming browser downloads and choosing the download folder, and lets users reorder pinned pages in the sidebar.

— Lists several browser UI additions with citation format named v0.1.905-beta

09 **RAG embedding model menu controls** NEW 45

Allows picking, pinning, and unloading RAG embedding models directly from the RAG menu.

— Gives a UI path but no further technical detail v0.1.905-beta

10 **Unified chat conversation and system prompt handling** IMPROVED 30

Trains prompt and completion message lists as one conversation, and includes the chat system prompt in chat exports and training data.

— Describes behavior without a concrete named surface v0.1.905-beta

└──▷ ALSO FROM THESE RELEASES

WAS THIS USEFUL?  

emisar

Feedback ⓘ v0.52.0 NOTES ↗ RANK

Emisar lets AI agents securely run infrastructure actions through MCP, with approval controls that protect production systems.

> Emisar v0.52.0 expands pack compatibility to newer MongoDB, Kubernetes, ClickHouse and Symbolicator versions, retires old Symbolicator packs, and reworks MFA/profile security UX plus pack reporting.

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

01 **Symbolicator pack retirement below 0.2.6** BREAKING 70

Updates the Symbolicator pack to `0.2.6`; versions earlier than `0.2.6` are retired and can no longer dispatch actions, requiring an upgrade.

– Clear before/after with exact version and required action v0.52.0

THINNER COVERAGE BELOW

02 **Pack compatibility for MongoDB, Kubernetes, ClickHouse, Symbolicator**

NEW 50

Adds pack compatibility for `MongoDB 8.3`, `Kubernetes 1.37`, `ClickHouse 26.9`, and `Symbolicator 26.9`, without dropping previously supported families.

– Names exact versions but no mechanism beyond compatibility v0.52.0

03 **MFA and profile security UX rework** IMPROVED 45

Reworks MFA setup and Profile security settings, including sign-out, enabled/disabled states, authenticator verification, and recovery-code flows.

– Lists sub-flows but no UI path or config detail v0.52.0

04 **Pack reporting shows runner-level content** IMPROVED 35

Pack reporting now shows which runners reported the exact pack content, removing repeated counts and empty-state tooltips.

– Describes behavior change but no navigation or metric detail v0.52.0

└──▷ **BREAKING ON UPGRADE**

! Symbolicator pack versions earlier than 0.2.6 are retired and cannot dispatch actions — upgrade to Symbolicator pack 0.2.6.

WAS THIS USEFUL? 👍 👎

◆ Local LLM Hardware

ASUS Ascent GX10

Feedback ⓘ price change · Oct 8, 2026

STORE ↗

ASUS's own GB10 Grace Blackwell desktop AI supercomputer — the only DGX Spark-class OEM twin with a scrapable live price, and it has climbed more than any other.

> Ascent GX10 · 128 GB / 1 TB `gx10-1tb`
>
> **$5,999** ~~$6,999~~ ▼ $1,000 · 14%
>
> list price at eshop.asus.com · Oct 8, 2026

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

Langfuse

Feedback ⓘ v4.55.0 NOTES ↗ RANK

Langfuse provides tracing, evaluation, and monitoring for LLM applications.

> Langfuse v4.55.0 adds batch score ingestion, Google Cloud Storage as a blob-storage export target, OpenAI decision models in evaluations, and several Skills workflow improvements including import, previews, and version diffing.

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

01 Batch score ingestion via POST /scores IMPROVED 80

The `POST /scores` endpoint now supports batch processing, allowing multiple scores to be submitted in a single API call instead of looping over individual requests — useful for bulk annotation pipelines.

Submit multiple scores in one request instead of looping over individual calls — useful for bulk annotation pipelines.

```
$ curl -X POST https://cloud.langfuse.com/api/public/scores \
  -H 'Authorization: Basic <base64-api-key>' \
  -H 'Content-Type: application/json' \
  -d '[{"traceId": "trace-abc", "name": "quality", "value": 0.9},
{"traceId": "trace-def", "name": "quality", "value": 0.4}]'
```

— Named endpoint with runnable curl example, but no limits given. v4.55.0

02 Google Cloud Storage export target NEW 70

Google Cloud Storage is now a supported export target for blob storage, supporting both service account key and Application Default Credentials (ADC) authentication; configurable via Settings > Blob Storage in the admin UI.

Configure Google Cloud Storage as a blob-storage export target using Application Default Credentials.

> 📍 In the Langfuse admin settings, navigate to Settings › Blob Storage and select Google Cloud Storage as the export target. Choose Application Default Credentials (ADC) for authentication and enter your GCS bucket name.

— Auth methods and UI path named, no API-level config shown. v4.55.0

THINNER COVERAGE BELOW

03 Skills import, previews, and version diffing NEW 45

Skills can now be imported from local files and ZIP archives, include rendered file previews with raw source access, and support side-by-side comparison of saved versions against draft changes.

— Names three constituent additions but no UI path or limits. v4.55.0

04 **Evaluator generation visibility as traces** IMPROVED 45

Each evaluator execution now publishes one generation, making evaluator LLM calls visible as traces.

— Explains the mechanism but no example of viewing it. v4.55.0

05 **OpenAI decision models in evaluations engine** NEW 35

The evaluations engine now supports OpenAI decision models as an evaluator type.

— Named capability but no setup detail or mechanism given. v4.55.0

06 **Independent gating of API key presets** IMPROVED 30

Project and organization presets for API keys are now gated independently.

— Thin description with no detail on prior behaviour or config. v4.55.0

07 **ClickHouse organization ID in debug panel** IMPROVED 25

The org debug information panel now shows the ClickHouse organization ID.

— Minor UI addition with minimal context given. v4.55.0

WAS THIS USEFUL?  

// END OF TRANSMISSION

The AI Toolchain · curated daily · October 8, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue