THE AI TOOLCHAIN

NO. 056

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED OCTOBER 10, 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 19 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 3 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.

A thin day, two unrelated items. Cloudflare Workers AI adds a multimodal decision model that scores audio, video and images ther instead of needing a separate classifier per modality, and

vim ? help

halves the hosted Clef-flash context window. Magpie makes gateway usage readable remotely with per-subagent attribution, so agent spend is attributable rather than a single opaque total.

> BUILD
>
> **Score audio, video and images in one call instead of stitching three classifiers together**
>
> The new multimodal decision model takes combined audio/video/image input, removing the glue code that normally fans a single asset out to per-modality models and reconciles their scores. Note the hosted Clef-flash context window is now nearly half its previous size, which matters if you were packing long prompts into it.
>
> Cloudflare Workers AI

> OPERATE
>
> **See which subagent burned the tokens, from somewhere other than the box running the gateway**
>
> Usage endpoints are now remote-readable and attribute spend to the subagent that caused it, so a multi-agent run stops reporting one aggregate number you cannot act on. Usage-limit resets and state-file recovery mean a crashed or throttled session resumes rather than needing a manual wipe.
>
> magpie

BUILD

◆ AI Coding Agents

yetone magpie

Feedback ⓘ v0.1.1166 CODE ↗ RANK

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

> Magpie added remote-readable gateway usage endpoints with subagent attribution, a full terminal UI and profile save/restore commands, `CODEX_HOME`-aware config routing for Codex, one-click routing-group templates, per-role thinking levels for omp, and new provider support for Bailian Token Plan, Qwen, and WorkBuddy AI, alongside usage-limit resets, state-file recovery, and assorted UI polish.

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

01 Gateway usage endpoints with subagent attribution NEW

90

Exposes `/v1/magpie/usage` and `/v1/magpie/usage/requests` gateway endpoints so usage data can be read from another machine using only a gateway key; Claude Code subagent calls now surface which subagent made the call and which subagent originally started it, visible in usage request data.

Pull the usage summary from a remote magpie gateway using only a gateway key, without needing direct access to the GUI.

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

Retrieve a paginated list of raw requests (including subagent attribution) from the gateway for audit or billing pipelines.

```
$ curl -H 'Authorization: Bearer <gateway-key>' 'https://<gateway-host>/v1/magpie/usage/requests?page=1'
```

— Exact endpoints with runnable curl examples v0.1.1165

02 Per-role thinking levels for omp agent roles NEW

90

Adds `plan`, `vision`, and `advisor` role fields to omp agent configuration, each accepting a model with an optional `:<level>` thinking-level suffix; every non-default role (subagent, small, slow, plan, vision, advisor) now supports its own independent

thinking level, stored as `<model>:<level>` in the role's config, e.g. `plan: anthropic/claude-opus-5:high` and `advisor: openai/gpt-6:low` in `~/.omp/agent/config.yml`.

Set a dedicated thinking level for omp's plan (Architect) and advisor roles so each uses a different reasoning depth without affecting the default model.

```yaml
defaultThinkingLevel: medium
modelRoles:
  default: anthropic/claude-opus-5
  plan: anthropic/claude-opus-5:high
  advisor: openai/gpt-6:low
```

— Exact config file, keys and syntax with runnable example `v0.1.1164`

---

03 **One-click routing group templates** NEW 83

Adds three routing group templates — `TemplateSmart`, `TemplateThrifty`, and `TemplateSteady` — that auto-build a routing group from the user's own models in one click: `smart` routes hard turns to the strongest model and easy turns to a cheap one; `thrifty` tries a cheap model first and escalates to the strongest on high-reasoning requests or failures; `steady` cascades across the strongest model of up to three providers for resilience.

— Full mechanism for each template, no exact command shown `v0.1.1163`

---

04 **Routing modes and per-conversation account affinity** NEW 81

Routing groups support `smart`, `order`, `rotate`, `usage`, and `pace` routing modes, with per-conversation account affinity to preserve vendor prompt-cache savings. A group can be built with `magpie group add "Opus anywhere" models=claude/claude-opus-5-5,copilot/claude-opus-5.5 routing=smart` and assigned via `magpie claude group/opus-anywhere` so a quota exhaustion on one account never stalls a session.

Fail over across two Claude subscriptions automatically so a quota exhaustion on one account never stalls your coding session.

```
$ magpie group add "Opus anywhere" models=claude/claude-opus-5-5,copilot/claude-opus-5.5 routing=smart
magpie claude group/opus-anywhere
```

— Named modes plus a runnable failover command `product docs`

05 **Plugin system with model-alias middleware** NEW 78

Plugins can be installed and configured via `magpie plugin add @magpie-community/middleware-model-map` and `magpie plugin options model-map '{"mapping": {"fast": "deepseek/deepseek-chat"}}'`, letting agents that hardcode a model name like `fast` be transparently redirected to a real provider model.

Install a model-alias middleware so agents that hardcode a model name like `fast` are transparently redirected to a real provider model.

```
$ magpie plugin add @magpie-community/middleware-model-map
magpie plugin options model-map '{"mapping": {"fast":
"deepseek/deepseek-chat"}}'
```

— Exact plugin commands given, but no broader plugin API detail `product docs`

06 **OpenAI-compatible local gateway** NEW 75

Any OpenAI-compatible agent can be pointed at magpie's local gateway to pick up all configured providers and routing groups without agent-specific config, via `export OPENAI_BASE_URL=http://127.0.0.1:3425/v1 OPENAI_API_KEY=magpie`.

Point any OpenAI-compatible agent at the local magpie gateway so it picks up all your configured providers and routing groups without agent-specific config.

```
$ export OPENAI_BASE_URL=http://127.0.0.1:3425/v1
OPENAI_API_KEY=magpie
```

— Exact env vars and port given, no further detail `product docs`

07 **CODEX_HOME-aware config routing for Codex** NEW 73

Supports the `CODEX_HOME` env var so a model picked for Codex is written to `$CODEX_HOME/config.toml`, with sign-in state read and switched from that location.

— Names env var and config path but no command example `v0.1.1166`

08 **Last-good state-file recovery** NEW 73

Corrupt or zeroed state files such as `logins.json` and `plugin-auth.json` now fall back to a `<file>.bak` copy, with the bad files archived as `<file>.bad-<time>`.

— Exact file names and fallback mechanism, no manual steps given `v0.1.1166`

09 **TUI and profile subcommands in CLI** NEW 60

Adds `magpie tui` to run the full magpie interface inside a terminal, plus `magpie save <name>` and `magpie use <name>` to save and restore named profiles that capture every agent's model setup.

— Named commands but no deeper mechanism given product docs

10 **Per-key usage limits with periodic and manual reset** IMPROVED 60

Adds per-gateway-key usage limits that reset every N days, with a Reset usage action that restarts the count from 0; a Codex usage reset now reflects immediately in the Providers editor and in routing without waiting for the next request's usage reading.

— Mechanism described, no exact config key for the interval v0.1.1164 v0.1.1166

THINNER COVERAGE BELOW

11 **WorkBuddy AI support** NEW 55

Adds WorkBuddy AI (international build) support, writing magpie's models to `~/.workbuddy-ai/models.json`.

— Names the config path but no further setup detail v0.1.1163

12 **Bailian Token Plan and Qwen provider presets** NEW 50

Adds Bailian Token Plan as its own provider preset with a dedicated decision model for routing groups, alongside a separate decision model for the Qwen AI platform.

— Names providers but no config keys or setup steps v0.1.1166

13 **Last-answered account visibility** NEW 40

The menu bar now shows the account that answered last, and the Usage page shows when each account last responded.

— UI change described, no navigation path given v0.1.1166

14 **Copy button for provider key rows** IMPROVED 40

Each provider key row now has its own Copy button in the Providers editor UI.

— Names the UI location but nothing further v0.1.1164

15 **Working-state detection for provider models** IMPROVED 40

Marks provider models as Working only after a successful test, clearing any picks that did not respond.

— Describes behavior change, no UI path or config given v0.1.1163

16 **Mobile layout wrapping fix** IMPROVED 35

On mobile, trailing row content wraps to its own line so provider, gateway key, model, call, and routed-request text is no longer truncated.

— Describes the fix but no navigation or config detail v0.1.1164

17 **OTLP export of billing data** NEW 25

Supports OTLP export of billing data to an external observability stack.

— No endpoint, format or config key given product docs

NEW INTEGRATION Google Vertex AI provider

v0.1.1162 CODE ↗ STABLE

Adds Google Vertex AI as a built-in preset with CLI, TUI, and GUI configuration and Application Default Credentials support.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.1.1162 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.1162
```

Adds `magpie provider add google-vertex project=…` CLI/TUI command to configure Google Vertex AI with location, credentials file, and service account as key-value pairs.

Adds Google Vertex AI as a built-in preset provider, serving Gemini models from a user's own Google Cloud project signed with Google credentials (OAuth, not an API key).

Adds support for Application Default Credentials discovery for Vertex AI: named credentials file, `GOOGLE_APPLICATION_CREDENTIALS`, or `gcloud auth application-default login` output, covering `authorized_user`, `service_account`, and `impersonated_service_account` kinds.

Adds Google Vertex AI provider UI editor with fields for Google Cloud project, location, credentials file, and service account.

Serves Claude Haiku 5.5 automatically alongside manually picked models once it appears on models.dev for accounts that have selected every model.

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Inference Providers & GPU Clouds

Baseten

Feedback ⓘ changelog · Oct 10, 2026 CHANGELOG ↗

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

NEW INTEGRATION DeepSeek V4.1 Flash Fast on Baseten Model APIs

WAS THIS USEFUL? 👍 👎

Cloudflare Workers AI

Feedback ⓘ changelog · Oct 10, 2026

CHANGELOG ↗ RANK

Cloudflare Workers AI runs open-weight models, including the Clef decision models, as serverless inference on Cloudflare's global network.

> Cloudflare Workers AI added a new multimodal decision model for combined audio/video/image scoring, while cutting the hosted Clef-flash context window nearly in half.

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

01 Clef-omni multimodal decision model NEW

85

Adds `@cf/cloudflare/clef-omni` for audio/video decisions, allowing a single Workers AI call to score mixed inputs — photos, audio clips, and video — against per-question prompts (e.g. checking if a label is visible, a unit sounds normal, or a fan is running) instead of chaining separate transcription and vision model calls.

Score a multimodal inspection — photo, audio clip, and video — with a single Workers AI call instead of chaining transcription and vision models.

javascript

```
const result = await env.AI.run('@cf/cloudflare/clef-omni', {
  model: 'clef-omni',
  prompt: 'Review the installation: a photo of the unit, an audio
recording of it running, and a video of the fan.',
  images: ['<base64-png>'],
  audio: ['data:audio/mpeg; base64,<base64-mp3>'],
  videos: ['data:video/mp4; base64,<base64-mp4>'],
  questions: {
    label_visible: { prompt: 'Is the model and serial number
label visible in the photo?' },
    sounds_normal: { prompt: 'Does the unit sound like it is
running smoothly, without rattling or grinding?' },
    fan_running:   { prompt: 'Is the fan running in the video?' }
  }
});
```

— Named model and runnable multimodal call shown, but internals unexplained

changelog-20261010-7f7d1b64

02 Clef-flash context window cut to 24K tokens BREAKING

75

The Cloudflare-hosted `Clef-flash` context window is reduced to 24K tokens, down from 64K; requests exceeding 24K input tokens must switch to `Clef` (which retains the 64K window) or self-host the model instead.

— Clear before/after numbers and migration options given, no timeline

changelog-20261010-7f7d1b64

└──▷ ALSO FROM THESE RELEASES

Block or log requests only when a specific detection (e.g. leaked credentials check) fails, without affecting requests where detections succeed.

```
$ # In a Custom Rule expression field:
any(cf.appsec.request.failed_detections[*] eq "waf_credential_check")
```

Enqueue 50 daily-report Workflow instances in one API call to avoid looping through individual create requests.

```javascript
const { created, errors } = await env.MY_WORKFLOW.createBatch({
  count: 50,
  params: { report: 'daily' }
});
for (const error of errors) {
  console.log(error.index, error.id, error.code, error.message);
}
```

└──▷ BREAKING ON UPGRADE

- ! Markdown for Agents converted responses no longer include the `x-markdown-tokens` or `x-original-tokens` headers; clients that use these values must calculate token counts themselves.
- ! Markdown for Agents converted responses no longer include `Content-Length`, as the Markdown body is now streamed.
- ! The Cloudflare-hosted `Clef-flash` context window is reduced to 24K tokens (from 64K); requests exceeding 24K input tokens must switch to Clef (64K window) or self-host.
- ! Passing an array directly to createBatch() is deprecated (the options-object form with `count` or `instances` replaces it); existing array-form code continues to work but should be migrated.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · October 10, 2026

Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue