Environment and debugging › **Debugging**

Environment and debugging

Debugging

This page covers debugging helpers for streaming output, especially when a provider mixes reasoning into normal text.

Runtime debug overrides

Use `/debug` in chat to set **runtime-only** config overrides (memory, not disk). `/debug` is disabled by default; enable with `commands.debug: true` . This is handy when you need to toggle obscure settings without editing `openclaw.json` .

Examples:

```
/debug show
/debug set messages.responsePrefix="[openclaw]"
/debug unset messages.responsePrefix
/debug reset
```

`/debug reset` clears all overrides and returns to the on-disk config.

Gateway watch mode

For fast iteration, run the gateway under the file watcher:

```
pnpm gateway:watch
```

This maps to:

```
node --watch-path src --watch-path tsconfig.json --watch-path package.
```

Add any gateway CLI flags after `gateway:watch` and they will be passed through on each restart.

Dev profile + dev gateway (—dev)

Use the dev profile to isolate state and spin up a safe, disposable setup for debugging. There are **two** `--dev` flags:

- **Global `--dev` (profile):** isolates state under `~/.openclaw-dev` and defaults the gateway port to `19001` (derived ports shift with it).
- `gateway --dev` **: tells the Gateway to auto-create a default config + workspace** when missing (and skip BOOTSTRAP.md).

Recommended flow (dev profile + dev bootstrap):

```
pnpm gateway:dev
OPENCLAW_PROFILE=dev openclaw tui
```

If you don't have a global install yet, run the CLI via `pnpm openclaw ...` .

What this does:

1. **Profile isolation** (global `--dev`)
   - `OPENCLAW_PROFILE=dev`
   - `OPENCLAW_STATE_DIR=~/.openclaw-dev`
   - `OPENCLAW_CONFIG_PATH=~/.openclaw-dev/openclaw.json`
   - `OPENCLAW_GATEWAY_PORT=19001` (browser/canvas shift accordingly)

2. **Dev bootstrap** (`gateway --dev`)
    - Writes a minimal config if missing (`gateway.mode=local` , bind loopback).
    - Sets `agent.workspace` to the dev workspace.
    - Sets `agent.skipBootstrap=true` (no BOOTSTRAP.md).
    - Seeds the workspace files if missing: `AGENTS.md` , `SOUL.md` , `TOOLS.md` , `IDENTITY.md` , `USER.md` , `HEARTBEAT.md` .
    - Default identity: **C3-PO** (protocol droid).
    - Skips channel providers in dev mode (`OPENCLAW_SKIP_CHANNELS=1`).

Reset flow (fresh start):

```
pnpm gateway:dev:reset
```

Note: `--dev` is a **global** profile flag and gets eaten by some runners. If you need to spell it out, use the env var form:

```
OPENCLAW_PROFILE=dev openclaw gateway --dev --reset
```

`--reset` wipes config, credentials, sessions, and the dev workspace (using `trash` , not `rm`), then recreates the default dev setup.

Tip: if a non-dev gateway is already running (launchd/systemd), stop it first:

```
openclaw gateway stop
```

Raw stream logging (OpenClaw)

OpenClaw can log the **raw assistant stream** before any filtering/formatting. This is the best way to see whether reasoning is arriving as plain text deltas (or as separate thinking blocks).

Enable it via CLI:

```
pnpm gateway:watch --raw-stream
```

Optional path override:

```
pnpm gateway:watch --raw-stream --raw-stream-path ~/.openclaw/logs/raw
```

Equivalent env vars:

```
OPENCLAW_RAW_STREAM=1
OPENCLAW_RAW_STREAM_PATH=~/.openclaw/logs/raw-stream.jsonl
```

Default file:

`~/.openclaw/logs/raw-stream.jsonl`

Raw chunk logging (pi-mono)

To capture **raw OpenAI-compat chunks** before they are parsed into blocks, pi-mono exposes a separate logger:

```
PI_RAW_STREAM=1
```

Optional path:

```
PI_RAW_STREAM_PATH=~/.pi-mono/logs/raw-openai-completions.jsonl
```

Default file:

```
~/.pi-mono/logs/raw-openai-completions.jsonl
```

> Note: this is only emitted by processes using pi-mono’s `openai-completions` provider.

Safety notes

Raw stream logs can include full prompts, tool output, and user data.

Keep logs local and delete them after debugging.

If you share logs, scrub secrets and PII first.

‹ Environment Variables Testing ›