



≡ Built-in tools > Thinking Levels

Built-in tools

Thinking Levels

What it does

Inline directive in any inbound body: `/t <level>` , `/think:<level>` , or `/thinking <level>` .

Levels (aliases): `off` | `minimal` | `low` | `medium` | `high` | `xhigh` (GPT-5.2 + Codex models only)

`minimal` → “think”

`low` → “think hard”

`medium` → “think harder”

`high` → “ultrathink” (max budget)

`xhigh` → “ultrathink+” (GPT-5.2 + Codex models only)

`x-high` , `x_high` , `extra-high` , `extra high` , and `extra_high` map to `xhigh` .

`highest` , `max` map to `high` .

Provider notes:

Z.AI (`zai/*`) only supports binary thinking (`on` / `off`). Any non-`off` level is treated as `on` (mapped to `low`).

Resolution order

1. Inline directive on the message (applies only to that message).

2. Session override (set by sending a directive-only message).
 3. Global default (`agents.defaults.thinkingDefault` in config).
 4. Fallback: low for reasoning-capable models; off otherwise.
-

Setting a session default

Send a message that is **only** the directive (whitespace allowed), e.g. `/think:medium` or `/t high`.

That sticks for the current session (per-sender by default); cleared by `/think:off` or session idle reset.

Confirmation reply is sent (`Thinking level set to high.` / `Thinking disabled.`). If the level is invalid (e.g. `/thinking big`), the command is rejected with a hint and the session state is left unchanged.

Send `/think` (or `/think:`) with no argument to see the current thinking level.

Application by agent

Embedded Pi: the resolved level is passed to the in-process Pi agent runtime.

Verbose directives (`/verbose` or `/v`)

Levels: `on` (minimal) | `full` | `off` (default).

Directive-only message toggles session verbose and replies `Verbose logging enabled.` / `Verbose logging disabled.` ; invalid levels return a hint without changing state.

`/verbose off` stores an explicit session override; clear it via the Sessions UI by choosing `inherit`.



Inline directive affects only that message; session/global defaults apply otherwise.

Send `/verbose` (or `/verbose:`) with no argument to see the current verbose level.

When verbose is on, agents that emit structured tool results (Pi, other JSON agents) send each tool call back as its own metadata-only message, prefixed with `<emoji> <tool-name>: <arg>` when available (path/command). These tool summaries are sent as soon as each tool starts (separate bubbles), not as streaming deltas.

When verbose is `full`, tool outputs are also forwarded after completion (separate bubble, truncated to a safe length). If you toggle `/verbose on|full|off` while a run is in-flight, subsequent tool bubbles honor the new setting.

Reasoning visibility (/reasoning)

Levels: `on|off|stream` .

Directive-only message toggles whether thinking blocks are shown in replies.

When enabled, reasoning is sent as a **separate message** prefixed with `Reasoning:` .

`stream` (Telegram only): streams reasoning into the Telegram draft bubble while the reply is generating, then sends the final answer without reasoning.

Alias: `/reason` .

Send `/reasoning` (or `/reasoning:`) with no argument to see the current reasoning level.

Related

Elevated mode docs live in [Elevated mode](#).

Heartbeats



Heartbeat probe body is the configured heartbeat prompt (default: `Read HEARTBEAT.md if it exists (workspace context). Follow it strictly. Do not infer or repeat old tasks from prior chats. If nothing needs attention, reply HEARTBEAT_OK.`). Inline directives in a heartbeat message apply as usual (but avoid changing session defaults from heartbeats).

Heartbeat delivery defaults to the final payload only. To also send the separate `Reasoning:` message (when available), set `agents.defaults.heartbeat.includeReasoning: true` or per-agent `agents.list[].heartbeat.includeReasoning: true` .

Web chat UI

The web chat thinking selector mirrors the session's stored level from the inbound session store/config when the page loads.

Picking another level applies only to the next message (`thinkingOnce`); after sending, the selector snaps back to the stored session level.

To change the session default, send a `/think:<level>` directive (as before); the selector will reflect it after the next reload.

< Elevated Mode

Reactions >

Powered by [mintlify](#)