Technical reference

Token Use and Costs

OpenClaw tracks **tokens**, not characters. Tokens are model-specific, but most OpenAI-style models average ~4 characters per token for English text.

How the system prompt is built

OpenClaw assembles its own system prompt on every run. It includes:

- Tool list + short descriptions
- Skills list (only metadata; instructions are loaded on demand with `read`)
- Self-update instructions
- Workspace + bootstrap files (`AGENTS.md`, `SOUL.md`, `TOOLS.md`, `IDENTITY.md`, `USER.md`, `HEARTBEAT.md`, `BOOTSTRAP.md` when new, plus `MEMORY.md` and/or `memory.md` when present). Large files are truncated by `agents.defaults.bootstrapMaxChars` (default: 20000), and total bootstrap injection is capped by `agents.defaults.bootstrapTotalMaxChars` (default: 150000). `memory/*.md` files are on-demand via memory tools and are not auto-injected.
- Time (UTC + user timezone)
- Reply tags + heartbeat behavior
- Runtime metadata (host/OS/model/thinking)

See the full breakdown in **System Prompt**.

What counts in the context window

Everything the model receives counts toward the context limit:

- System prompt (all sections listed above)
- Conversation history (user + assistant messages)
- Tool calls and tool results
- Attachments/transcripts (images, audio, files)
- Compaction summaries and pruning artifacts
- Provider wrappers or safety headers (not visible, but still counted)

For images, OpenClaw downscales transcript/tool image payloads before provider calls. Use `agents.defaults.imageMaxDimensionPx` (default: `1200`) to tune this:

- Lower values usually reduce vision-token usage and payload size.
- Higher values preserve more visual detail for OCR/UI-heavy screenshots.

For a practical breakdown (per injected file, tools, skills, and system prompt size), use `/context list` or `/context detail`. See **Context**.

How to see current token usage

Use these in chat:

- `/status` → **emoji-rich status card** with the session model, context usage, last response input/output tokens, and **estimated cost** (API key only).
- `/usage off|tokens|full` → appends a **per-response usage footer** to every reply.
  - Persists per session (stored as `responseUsage`).

OAuth auth **hides cost** (tokens only).

`/usage cost` → shows a local cost summary from OpenClaw session logs.

Other surfaces:

TUI/Web TUI: `/status` + `/usage` are supported.

CLI: `openclaw status --usage` and `openclaw channels list` show provider quota windows (not per-response costs).

Cost estimation (when shown)

Costs are estimated from your model pricing config:

```
models.providers.<provider>.models[].cost
```

These are **USD per 1M tokens** for `input`, `output`, `cacheRead`, and `cacheWrite`. If pricing is missing, OpenClaw shows tokens only. OAuth tokens never show dollar cost.

Cache TTL and pruning impact

Provider prompt caching only applies within the cache TTL window. OpenClaw can optionally run **cache-ttl pruning**: it prunes the session once the cache TTL has expired, then resets the cache window so subsequent requests can re-use the freshly cached context instead of re-caching the full history. This keeps cache write costs lower when a session goes idle past the TTL.

Configure it in and see the behavior details in .

Heartbeat can keep the cache **warm** across idle gaps. If your model cache TTL is `1h`, setting the heartbeat interval just under that (e.g., `55m`) can avoid re-caching the full prompt, reducing cache write costs.

For Anthropic API pricing, cache reads are significantly cheaper than input tokens, while cache writes are billed at a higher multiplier. See Anthropic's prompt caching pricing for the latest rates and TTL multipliers: **https://docs.anthropic.com/docs/build-with-claude/prompt-caching**

Example: keep 1h cache warm with heartbeat

```yaml
agents:
  defaults:
    model:
      primary: "anthropic/claude-opus-4-6"
    models:
      "anthropic/claude-opus-4-6":
        params:
          cacheRetention: "long"
    heartbeat:
      every: "55m"
```

Example: enable Anthropic 1M context beta header

Anthropic's 1M context window is currently beta-gated. OpenClaw can inject the required `anthropic-beta` value when you enable `context1m` on supported Opus or Sonnet models.

```yaml
agents:
  defaults:
    models:
      "anthropic/claude-opus-4-6":
        params:
          context1m: true
```

This maps to Anthropic's `context-1m-2025-08-07` beta header.

Tips for reducing token pressure

- Use `/compact` to summarize long sessions.
- Trim large tool outputs in your workflows.
- Lower `agents.defaults.imageMaxDimensionPx` for screenshot-heavy sessions.
- Keep skill descriptions short (skill list is injected into the prompt).
- Prefer smaller models for verbose, exploratory work.

See for the exact skill list overhead formula.

‹ Wizard Reference grammY ›