Fundamentals

Context

"Context" is **everything OpenClaw sends to the model for a run**. It is bounded by the model's **context window** (token limit).

Beginner mental model:

- **System prompt** (OpenClaw-built): rules, tools, skills list, time/runtime, and injected workspace files.
- **Conversation history**: your messages + the assistant's messages for this session.
- **Tool calls/results + attachments**: command output, file reads, images/audio, etc.

Context is *not the same thing* as "memory": memory can be stored on disk and reloaded later; context is what's inside the model's current window.

Quick start (inspect context)

- `/status` → quick "how full is my window?" view + session settings.
- `/context list` → what's injected + rough sizes (per file + totals).
- `/context detail` → deeper breakdown: per-file, per-tool schema sizes, per-skill entry sizes, and system prompt size.
- `/usage tokens` → append per-reply usage footer to normal replies.

`/compact` → summarize older history into a compact entry to free window space.

See also: Slash commands, Token use & costs, Compaction.

Example output

Values vary by model, provider, tool policy, and what's in your workspace.

/context list

```
🧠 Context breakdown
Workspace: <workspaceDir>
Bootstrap max/file: 20,000 chars
Sandbox: mode=non-main sandboxed=false
System prompt (run): 38,412 chars (~9,603 tok) (Project Context 23,901 chars (~5,

Injected workspace files:
- AGENTS.md: OK | raw 1,742 chars (~436 tok) | injected 1,742 chars (~436 tok)
- SOUL.md: OK | raw 912 chars (~228 tok) | injected 912 chars (~228 tok)
- TOOLS.md: TRUNCATED | raw 54,210 chars (~13,553 tok) | injected 20,962 chars (~
- IDENTITY.md: OK | raw 211 chars (~53 tok) | injected 211 chars (~53 tok)
- USER.md: OK | raw 388 chars (~97 tok) | injected 388 chars (~97 tok)
- HEARTBEAT.md: MISSING | raw 0 | injected 0
- BOOTSTRAP.md: OK | raw 0 chars (~0 tok) | injected 0 chars (~0 tok)

Skills list (system prompt text): 2,184 chars (~546 tok) (12 skills)
Tools: read, edit, write, exec, process, browser, message, sessions_send, …
Tool list (system prompt text): 1,032 chars (~258 tok)
Tool schemas (JSON): 31,988 chars (~7,997 tok) (counts toward context; not shown
Tools: (same as above)

Session tokens (cached): 14,250 total / ctx=32,000
```

/context detail

```
🧠 Context breakdown (detailed)
…
Top skills (prompt entry size):
- frontend-design: 412 chars (~103 tok)
- oracle: 401 chars (~101 tok)
… (+10 more skills)

Top tools (schema size):
- browser: 9,812 chars (~2,453 tok)
- exec: 6,240 chars (~1,560 tok)
… (+N more tools)
```

What counts toward the context window

Everything the model receives counts, including:

- System prompt (all sections).
- Conversation history.
- Tool calls + tool results.
- Attachments/transcripts (images/audio/files).
- Compaction summaries and pruning artifacts.
- Provider “wrappers” or hidden headers (not visible, still counted).

How OpenClaw builds the system prompt

The system prompt is **OpenClaw-owned** and rebuilt each run. It includes:

- Tool list + short descriptions.
- Skills list (metadata only; see below).
- Workspace location.
- Time (UTC + converted user time if configured).

Runtime metadata (host/OS/model/thinking).

Injected workspace bootstrap files under **Project Context**.

Full breakdown: **System Prompt**.

Injected workspace files (Project Context)

By default, OpenClaw injects a fixed set of workspace files (if present):

- `AGENTS.md`
- `SOUL.md`
- `TOOLS.md`
- `IDENTITY.md`
- `USER.md`
- `HEARTBEAT.md`
- `BOOTSTRAP.md` (first-run only)

Large files are truncated per-file using `agents.defaults.bootstrapMaxChars` (default `20000` chars). OpenClaw also enforces a total bootstrap injection cap across files with `agents.defaults.bootstrapTotalMaxChars` (default `150000` chars). `/context` shows **raw vs injected** sizes and whether truncation happened.

Skills: what's injected vs loaded on-demand

The system prompt includes a compact **skills list** (name + description + location). This list has real overhead.

Skill instructions are *not* included by default. The model is expected to `read` the skill's `SKILL.md` **only when needed**.

Tools: there are two costs

Tools affect context in two ways:

1. **Tool list text** in the system prompt (what you see as “Tooling”).
2. **Tool schemas** (JSON). These are sent to the model so it can call tools. They count toward context even though you don’t see them as plain text.

`/context detail` breaks down the biggest tool schemas so you can see what dominates.

Commands, directives, and “inline shortcuts”

Slash commands are handled by the Gateway. There are a few different behaviors:

Standalone commands: a message that is only `/...` runs as a command.

Directives: `/think`, `/verbose`, `/reasoning`, `/elevated`, `/model`, `/queue` are stripped before the model sees the message.

Directive-only messages persist session settings.

Inline directives in a normal message act as per-message hints.

Inline shortcuts (allowlisted senders only): certain `/...` tokens inside a normal message can run immediately (example: “hey /status”), and are stripped before the model sees the remaining text.

Details: **Slash commands**.

Sessions, compaction, and pruning (what persists)

What persists across messages depends on the mechanism:

- **Normal history** persists in the session transcript until compacted/pruned by policy.
- **Compaction** persists a summary into the transcript and keeps recent messages intact.
- **Pruning** removes old tool results from the *in-memory* prompt for a run, but does not rewrite the transcript.

Docs: **Session**, **Compaction**, **Session pruning**.

What `/context` actually reports

`/context` prefers the latest **run-built** system prompt report when available:

- `System prompt (run)` = captured from the last embedded (tool-capable) run and persisted in the session store.
- `System prompt (estimate)` = computed on the fly when no run report exists (or when running via a CLI backend that doesn't generate the report).

Either way, it reports sizes and top contributors; it does **not** dump the full system prompt or tool schemas.

‹ System Prompt Agent Workspace ›