



≡ Fundamentals > System Prompt

Fundamentals

System Prompt

OpenClaw builds a custom system prompt for every agent run. The prompt is **OpenClaw-owned** and does not use the pi-coding-agent default prompt.

The prompt is assembled by OpenClaw and injected into each agent run.

Structure

The prompt is intentionally compact and uses fixed sections:

Tooling: current tool list + short descriptions.

Safety: short guardrail reminder to avoid power-seeking behavior or bypassing oversight.

Skills (when available): tells the model how to load skill instructions on demand.

OpenClaw Self-Update: how to run `config.apply` and `update.run` .

Workspace: working directory (`agents.defaults.workspace`).

Documentation: local path to OpenClaw docs (repo or npm package) and when to read them.

Workspace Files (injected): indicates bootstrap files are included below.

Sandbox (when enabled): indicates sandboxed runtime, sandbox paths, and whether elevated exec is available.

Current Date & Time: user-local time, timezone, and time format.



Reply Tags: optional reply tag syntax for supported providers.

Heartbeats: heartbeat prompt and ack behavior.

Runtime: host, OS, node, model, repo root (when detected), thinking level (one line).

Reasoning: current visibility level + /reasoning toggle hint.

Safety guardrails in the system prompt are advisory. They guide model behavior but do not enforce policy. Use tool policy, exec approvals, sandboxing, and channel allowlists for hard enforcement; operators can disable these by design.

Prompt modes

OpenClaw can render smaller system prompts for sub-agents. The runtime sets a `promptMode` for each run (not a user-facing config):

`full` (default): includes all sections above.

`minimal` : used for sub-agents; omits **Skills**, **Memory Recall**, **OpenClaw Self-Update**, **Model Aliases**, **User Identity**, **Reply Tags**, **Messaging**, **Silent Replies**, and **Heartbeats**. Tooling, **Safety**, **Workspace**, **Sandbox**, **Current Date & Time** (when known), **Runtime**, and injected context stay available.

`none` : returns only the base identity line.

When `promptMode=minimal` , extra injected prompts are labeled **Subagent Context** instead of **Group Chat Context**.

Workspace bootstrap injection

Bootstrap files are trimmed and appended under **Project Context** so the model sees identity and profile context without needing explicit reads:

`AGENTS.md``SOUL.md``TOOLS.md` >`IDENTITY.md``USER.md``HEARTBEAT.md``BOOTSTRAP.md` (only on brand-new workspaces)`MEMORY.md` and/or `memory.md` (when present in the workspace; either or both may be injected)

All of these files are **injected into the context window** on every turn, which means they consume tokens. Keep them concise – especially `MEMORY.md`, which can grow over time and lead to unexpectedly high context usage and more frequent compaction.

Note: `memory/*.md` daily files are **not** injected automatically. They are accessed on demand via the `memory_search` and `memory_get` tools, so they do not count against the context window unless the model explicitly reads them.

Large files are truncated with a marker. The max per-file size is controlled by `agents.defaults.bootstrapMaxChars` (default: 20000). Total injected bootstrap content across files is capped by `agents.defaults.bootstrapTotalMaxChars` (default: 150000). Missing files inject a short missing-file marker.

Sub-agent sessions only inject `AGENTS.md` and `TOOLS.md` (other bootstrap files are filtered out to keep the sub-agent context small).

Internal hooks can intercept this step via `agent:bootstrap` to mutate or replace the injected bootstrap files (for example swapping `SOUL.md` for an alternate persona).

To inspect how much each injected file contributes (raw vs injected, truncation, plus tool schema overhead), use `/context list` or `/context detail`. See [Context](#).

>

Time handling

The system prompt includes a dedicated **Current Date & Time** section when the user timezone is known. To keep the prompt cache-stable, it now only includes the **time zone** (no dynamic clock or time format).

Use `session_status` when the agent needs the current time; the status card includes a timestamp line.

Configure with:

```
agents.defaults.userTimezone
```


```
agents.defaults.timeFormat ( auto | 12 | 24 )
```

See [Date & Time](#) for full behavior details.

Skills

When eligible skills exist, OpenClaw injects a compact **available skills list** (`formatSkillsForPrompt`) that includes the **file path** for each skill. The prompt instructs the model to use `read` to load the `SKILL.md` at the listed location (workspace, managed, or bundled). If no skills are eligible, the Skills section is omitted.

```
<available_skills>
  <skill>
    <name>...</name>
    <description>...</description>
    <location>...</location>
  </skill>
</available_skills>
```

This keeps the base prompt small while still enabling targeted skill
age.

>

Documentation

When available, the system prompt includes a **Documentation** section that points to the local OpenClaw docs directory (either `docs/` in the repo workspace or the bundled npm package docs) and also notes the public mirror, source repo, community Discord, and ClawHub (<https://clawhub.com>) for skills discovery. The prompt instructs the model to consult local docs first for OpenClaw behavior, commands, configuration, or architecture, and to run `openclaw status` itself when possible (asking the user only when it lacks access).

< Agent Loop

Context >

Powered by mintlify