NAME

cmagent − tool-using agent on a local language model

SYNOPSIS

cmagent [*OPTION*]... *MODEL* [*TASK*]
cmagent [*OPTION*]... **−−api−base** *URL* **−−api−model** *NAME* [*TASK*]

DESCRIPTION

cmagent runs a conversation in which the model may call tools: read, list and search files, query git and a language server, and, if enabled, edit files and run shell commands in a sandbox. Each tool result is appended to the conversation as a tool turn, and the model continues until it answers in prose.

MODEL is any model **cmgenerate**(1) loads. A tool call is constrained while it is sampled to valid JSON naming a known tool. *TASK* is the first task; further tasks are read at the prompt.

With **−−api−base** the model is instead a remote OpenAI-compatible service (FhGenie, vLLM, llama.cpp, OpenAI, or **cmserve**(1) itself): no weights are loaded, the conversation is sent on each turn to *URL*/chat/completions, and tools are offered both as the OpenAI **tools** schema and as ChatML tags in the system turn (the output cannot be constrained as a local model's is). A model trained for function-calling returns structured calls; a weaker one may follow the tags, or neither. The window, prefill cache, speculation and image input do not apply, and context use is estimated.

OPTIONS

Generation

−c, **−−context** *N*
: Context length to allocate. Default: 32768.

−−window *N*, **−−sinks** *N*
: Sliding-window attention; see **cmgenerate**(1).

−n, **−−steps** *N*
: Maximum tokens per answer. Default: until the end of the turn.

−t, **−k**, **−p**, **−s**
: Temperature, top-k, top-p, seed; as in **cmgenerate**(1).

−−mmproj *PATH*
: The vision tower for a task's images, an mmproj GGUF or a HuggingFace checkout; see **IMAGES** and **cmgenerate**(1). Default: the model's checkout itself, else an *mmproj*.gguf* beside the model.

−−system *TEXT*
: System prompt, placed before the tool declarations.

−−speculate *N*, **−−no−mtp**, **−−mtp−from** *PATH*
: Speculative decoding; as in **cmgenerate**(1).

−−effort *LEVEL*
: Reasoning effort; as in **cmgenerate**(1).

Tools

−−write
: Enable the file writing tools: **write_file**, **edit_file**, **patch_file**, **move_file**, **delete_file**, **undo_edit**, **changes**, **run_check**.

−−run
: **−−shell offline**: **run_command** (shell commands in the sandbox) and **check_command** (background commands).

−−yes Do not ask for approval before writing or running. Stored refusals still apply.

−−no−check
: Do not run the project check after an edit.

−−hops *N*
Maximum tool calls per task. Default: unlimited. The third identical call that acts is refused, as is a repeated reading call whose result has not changed.

−−free−calls
Do not constrain tool calls to valid JSON and known tool names.

−−command−timeout *S*
Maximum seconds a command may request. Default: 3600. A command that sets no timeout and does not run in the background gets 300.

−−task−timeout *S*
Maximum seconds per task. Default: 14400.

−−max−output *BYTES*
Tool output kept in the context per result. Default: 16384, less if the context is small. The full output is written to */tmp/output−N.txt* in the sandbox.

Sandbox

−−root *DIR*
File tools do not read or write outside *DIR*. Default: the working directory.

−−allow−read *PATH*
Extra read-only path for the shell and the reading tools. Repeatable.

−−allow−write *PATH*
Extra writable path for the shell. Repeatable.

−−shell *SANDBOX*
Offer **run_command** and **check_command**, whose commands run in *SANDBOX*: **off** (no shell, the default), **offline** (own network namespace, loopback only), **online** (the host's network), or a wrapper from *~/.config/cmagent/sandboxes*; see **SANDBOX**.

−−list−sandboxes
Print the sandboxes as JSON, and exit.

−−allow−network
Old: **−−shell online** with **−−run**, and **−−web untrusted** unless **−−web** is given.

−−web *LEVEL*
Offer the web tools **fetch_url**, **web_search** and **news_search** for the providers at or below *LEVEL*: **off**, **local**, **trusted** or **untrusted**. Default: **off**. See **WEB PROVIDERS**.

−−web−ask *LEVEL*
A web call to a provider at or above *LEVEL* asks for approval first, like a tool that acts; **never** for none (the default). An approval can be kept for the exact address or query, every call, or every address on the same site.

−−allow−fetch−url
−−web untrusted.

−−no−fetch−url
−−web off, whatever else is given.

−−ask−web
−−web−ask local.

−−providers *FILE*
Read providers from *FILE* instead of *~/.config/cmagent/providers*.

−−fetch *MODE*
any (default): **fetch_url** reaches any public address; **listed**: only the hosts of providers.

−−list−providers
Print the providers of **−−root** and the user, built-in ones last, as JSON, and exit.

−−no−gpu−in−shell
Hide the GPU device nodes from shell commands.

−−no−home−in−shell
Hide the toolchains under *$HOME* from shell commands.

Session

−−once
Run *TASK* and exit.

−−require−tool
With **−−once**, exit with an error unless a tool call succeeded.

−−resume
Restore the saved conversation and tool results, without replaying actions.

−−session−id *ID*
Keep a separate transcript within this project.

−−no−prefix−cache
Recompute the system turn instead of loading its KV cache from *~/.cache/cmagent*. A hybrid model's file also holds its recurrent state at the last anchor before the user's text, and is used only whole; a recurrent state on the GPU is not saved.

−−log *PATH*
Append tool calls and results to *PATH* as JSON lines.

−q, **−−quiet**
No load report; warnings are still printed.

−−vi vi key bindings at the prompt.

−−changes
Print what the last task of the session under **−−root** (**−−session−id**) changed, as JSON with a diff against its checkpoint, and exit. Reads only; needs no model.

−−tui Full-screen interface: transcript, status line and input, with approvals answered in place. Runs the agent as a child process over **−−bridge**.

−−bridge
JSON Lines front-end protocol on standard input and standard error. **{"type":"undo","what":"action"}** (or **"task"**) puts files back as **/undo** does and answers with an **undone** event. **{"type":"retry","text":"...","undo":false}** takes the last task out of the conversation (and with **undo** puts its files back first) and runs **text** in its place; after a compaction during that task, from the transcript kept from before it. Refused, with a failed **done**, when no task ran. **{"type":"cancel"}** stops the running task as Ctrl-C does.

Placement

−−gpu, **−−gpu−backend**, **−−gpu−device**, **−−gpu−split**, **−−gpu−as**, **−−cpu−as**, **−−kv−format**, **−−lora**, **−−lora−as**: as in **cmgenerate**(1).

Remote

−−api−base *URL*
Chat-completions base, e.g. *https://host/v1*; */chat/completions* is appended unless already present. Its presence selects the remote backend.

−−api−model *NAME*
The model the service knows, sent as *model* and shown in the report.

−−api−level *LEVEL*
: The service's trust level: **local**, **trusted** or **untrusted**. Default: the level of a **model** line in a providers file that covers **−−api−base**, else **local** at a loopback address, else **untrusted**.

−−model−level *LEVEL*
: Refuse to start with a model service above *LEVEL* (default: **untrusted**). The service receives the whole conversation, files included. Local weights are **local**.

−−api−key−env *VAR*
: Environment variable holding the API key (default: **CMAGENT_API_KEY**, then **OPENAI_API_KEY**). The key is read from the environment, never an argument, so it stays out of the process list. A service that needs no key is fine.

−−api−key−header *H*
: The header the key is sent in (default: **Authorization**, as **Bearer** *KEY*). Any other name sends the bare key, for the **api−key** header Azure and FhGenie gateways use.

−−api−header *H*
: An extra "*Name: Value*" header, repeatable — for instance an **api−version** an Azure endpoint wants.

−−api−no−tools
: Do not send the OpenAI **tools** schema. By default it is sent so a model trained for function-calling returns structured calls, which are turned into the same tags the ChatML system turn asks for; a service that ignores the field falls back to those tags. This flag keeps only the tags.

TOOLS

Always offered: **list_directory** and **find_files** (discovery), **read_file** (numbered lines; a PDF as its text; an image as the picture, see **IMAGES**), **search_files** (ripgrep if installed; **regex** and **context** arguments), **outline** (symbols with line ranges, from the language server or declaration-like lines), **git** (status, diff, log, show, blame; read-only, where there is a *.git*), **remember** and **recall** (the model's notes), and **delegate** (a sub-task run in the same context window and rewound when it answers; at most 40 calls, no nested delegation).

With **−−web** above **off**, each web tool whose service has an allowed provider: **fetch_url** (a page over http or https as text: title, headings, lists and table rows kept, markup, scripts and menus dropped; **links** keeps links as **[text](address)**). It runs **curl**(1) with up to 5 redirects, 30 seconds and 4 MiB a request; every hop goes to an allowed provider. Each host is resolved and pinned with **−−resolve**; loopback, private, link-local and otherwise reserved addresses are refused except for a local or trusted provider from a providers file. **web_search** (title, address and snippet of up to 10 results from the search provider, by default *https://html.duckduckgo.com/html/*, ads left out; **region** such as de−de) and **news_search** (the newest articles of the last 1 to 30 days from *https://newsradio.creative−memory.de/2/*, English and German sources: title, source, time, category, summary and address; every word must occur, **OR** for either, quotes for a phrase, a trailing ***** for a prefix; optional **category**). **web_search** and **news_search** take an optional **provider**. Results name the provider and its level. With these the system turn states the date and lists the allowed providers.

WEB PROVIDERS

A provider is where a web tool's request goes, with a trust level by where the data ends up: **local** (this machine or the institution), **trusted** (a service under contract or own control), **untrusted** (the public internet). A self-hosted SearXNG that forwards to public engines is untrusted; one with only local engines is local.

Built in: **duckduckgo** (search, untrusted), **newsradio** (news, trusted), **web** (fetch, any public address, untrusted). More in *~/.config/cmagent/providers* or the file **−−providers** names, never in a project's *.cmagent/*: a level, and a private address, are the user's to grant. One a line:

```
# NAME    SERVICE LEVEL   ADAPTER    ENDPOINT
index     search  local   searxng    http://10.0.0.5:8888/search
intra     fetch   local   pages      https://intranet.example.org/
newsradio news    local   newsradio  cmd:news-browse /srv/news/index
```

An endpoint **cmd:PROGRAM ARGS** (split at spaces) runs a local program instead of an HTTP request: the parameters arrive as one JSON object on stdin, the answer is read from stdout as the HTTP service would send

it. newsradio's **news−browse** answers the browse API so from a local index.

SERVICE is **search**, **news**, **fetch** or **model** (an **−−api−base** service, adapter **openai**, giving its level; e.g. **fhgenie model trusted openai https://fhgenie−preview.c.fraunhofer.de/v1**); ADAPTER **duckduckgo** or **searxng** for search (SearXNG needs **json** under **search.formats**), **newsradio** for news, **pages** for fetch. The first of a name wins (the file, then built-in), so a line can replace a built-in provider; within a service the first allowed one is the default. A provider covers addresses with its endpoint's scheme, host, port and a path under its path; a fetch to another public address goes to **web**. A host written ***.HOST** covers every subdomain too, e.g. **fhg fetch trusted pages https://*.fraunhofer.de/**. With **−−fetch listed**, **web** is left out and **fetch_url** reaches only the hosts of providers: a whitelist.

A language server, the command in *.cmagent/lsp* or else **rust−analyzer** in a Cargo project, is started in the sandbox and offered as **find_definition**, **find_references**, **describe_symbol** and **find_symbol**.

APPROVALS

Writing and running show the edit preview or command and ask, unless **−−yes** is given. An answer can be kept for the exact command or path, for every command of that program or every path of that tool, or for a pattern (**cargo test ***, **src/*.rs**), for the session or for the project (*.cmagent/allowed.json*). A command is covered only if every **&&**, **|** and **;** segment is; one containing a **$(...)** substitution only by an exact answer.

A refusal can be stored the same way (**/never rm**, **/never git push ***, or **!** before the answer letter). A refusal takes precedence over every approval, over **−−yes** and over an interactive yes; one matching segment is enough.

SANDBOX

Shell commands require Linux and bubblewrap. The project is writable, *.git* included; system directories are read-only; toolchain caches under *$HOME* (rustup, cargo registry, npm, pip) are read-only and credential files are omitted; the GPU is visible; network access is disabled. */tmp* is *.cmagent/scratch* and persists between commands; **read_file** can read it. Agent state under *.cmagent* is not writable. All commands of a session share one network namespace, so a server started in the background is reachable at 127.0.0.1 by later commands but not from the host.

A command that needs the GPU's memory sets **gpu: true**; on Vulkan the agent's model is then moved off the device for the duration and restored before the model reads the result.

Ctrl-C terminates running commands and their background children.

−−shell names the sandbox. **offline** and **online** are the bubblewrap sandbox above, without and with the host's network. *~/.config/cmagent/sandboxes* adds wrappers, one a line: *NAME NETWORK COMMAND...*, NETWORK **none** or **net** as declared, words split as a shell would ('...', "...", # comments). Each command line is given to COMMAND as its last argument, from the project root; **{root}** in COMMAND is replaced by it. Only PATH, HOME, USER, LOGNAME, LANG, TERM, XDG_RUNTIME_DIR, SSH_AUTH_SOCK and DBUS_SESSION_BUS_ADDRESS are passed on:

```
rust none podman run −−rm −i −−network=none −v {root}:{root} −w {root} docker.io/libra
```

A wrapper runs on the host outside bubblewrap, so it is read from the user's file only, not from a project's *.cmagent/*, and the built-in names cannot be replaced. What commands see inside it (files, toolchains, the GPU) is the wrapper's matter.

PROJECT CHECK

After every edit the project check runs in the sandbox and its first diagnostics follow the edit's result: *.cmagent/check* if present, otherwise **cargo check −−all−targets −q −−message−format=short** for a *Cargo.toml* or **go vet ./...** for a *go.mod*.

HOOKS

Executables in *.cmagent/hooks* run at three points:

after−edit
: Instead of the project check.

before−run
: Receives each command line on standard input. A non-zero exit refuses the command; its output is the reason given to the model.

after−task
: After a task that performed an action is answered. A failure is reported to the model once.

PROJECT TOOLS

An executable in *.cmagent/tools* is offered as a tool under its file name. Invoked with **−−describe**, it prints a JSON object

```
{"about": ..., "acts": bool,
 "takes": [{"name": ..., "about": ..., "kind": ..., "required": bool}]}
```

with **kind** one of string, integer, boolean. A call runs it in the sandbox with the arguments as one JSON object on standard input; its standard output is the result and a non-zero exit an error. A tool with **acts** true is offered only with **−−write** or **−−run** and asks for approval like the built-in ones.

SKILLS

A directory holding a *SKILL.md* under *.cmagent/skills* (the project's) or *~/.config/cmagent/skills* (*$XDG_CONFIG_HOME/cmagent/skills*; the user's) is a skill; the project's wins on a shared name. Front matter

```
---
name: release
description: Cut a release: bump the version, tag, write the notes.
---
```

gives its name (else the directory's) and description (else the first line that is not a heading). The system turn lists every skill's name and description (at most 50 skills, a description cut at 1024 characters) and offers **use_skill**, which returns the rest of *SKILL.md* with the other files of the skill's directory listed, or, with **file**, one of those files. A skill written for Claude Code (the Agent Skills format) works as it is: copy its directory there. **−−list−skills** prints them as JSON (name, description, directory, scope) and exits without loading a model.

IMAGES

A task's lines **Attached file:** *PATH* that name an image (PNG, JPEG, WebP, GIF, BMP) inside the workspace go to the model with the task's first turn, where it reads images (Qwen3.5 and Qwen3-VL GGUFs with their *mmproj*); the line stays in the text. A model that does not read images notes each one as not read. The web UI writes one such line per upload. **read_file** of an image puts the picture in its result for a model that reads images, and fails with "this model does not read images" otherwise. A resumed session reads its images again from their files; one whose file is gone is lost. Compaction drops images with the turns it summarises.

INTERACTIVE COMMANDS

/help List the commands.

/new, **/system** *TEXT*
: Start the conversation over, keeping the model; with new standing instructions.

/temperature, **/top−k**, **/top−p**, **/seed**, **/steps**
: Set the sampling option; **/settings** shows the current values.

/context, **/tokens** *TEXT*
: Context window use; tokenisation of *TEXT*.

/undo, **/undo−task**
: Restore the last action, or the whole task; conflicting edits are refused.

/changes, **/recover**
: Show the task's changes; review and undo interrupted work.

/remember *TEXT*, **/memory**
: Pin a decision; show the pins and the model's notes.

/compact
: Archive the transcript and replace it with a summary; also automatic near the context limit. Goals and pins are kept verbatim.

/allowed, **/forget** *N*, **/never** *PATTERN*
: List stored approvals and refusals, drop one, store a refusal.

/tools, **/jobs**
: Offered tools; background commands.

/last, **/verbose**
: Show the latest tool result in full; toggle showing every result in full.

/quit Exit.

/edit Write the next task in *$EDITOR*.

!*command*
: Run *command* in your own shell, outside the sandbox; **!!***command* also keeps its output for the next task.

A line typed while the model works is inserted with the next tool result, or becomes the next task if the model has answered. Tab completes commands and paths. Ctrl-C stops the task and keeps the model loaded; Ctrl-D exits.

FILES

~/.config/cmagent/config, *.cmagent/config*
: Options, one per line with its value after it; # starts a comment; a bare line names the model. Read in this order; the command line overrides both.

.cmagent/skills//SKILL.md*, *~/.config/cmagent/skills/*/SKILL.md*
: Skills; see **SKILLS**.

~/.config/cmagent/providers
: Web and model providers; see **WEB PROVIDERS**.

AGENTS.md
: Project notes, included in the system turn.

.cmagent/notes.md
: The model's notes; the latest 3 KiB open every task.

.cmagent/allowed.json
: Stored approvals and refusals for the project.

.cmagent/history
: Prompt history.

.cmagent/
: Sessions, checkpoints (source files up to 64 MiB) and action evidence.

~/.cache/cmagent/
: KV cache of the system turn, one file per model and project.

SEE ALSO

cmgenerate(1), **cmfinetune**(1), **bwrap**(1)