

Every command, flag and shortcut in Claude Code 2.1.259.

208 entries checked 3 September 2026 read from the program, not typed by hand codasmedia.com/claude-code-commands

Session & context20	
<code>/add-dir</code>	Add a new working directory
<code>/autocompact</code>	Configure the auto-compact window size GATED
<code>/branch</code>	Create a branch of the current conversation at this point
<code>/brief</code>	Toggle brief-only mode
<code>/btw</code>	Ask a quick side question without interrupting the main conversation
<code>/cd</code>	Move this session to a new working directory
<code>/clear</code>	Start a new session with empty context; previous session stays on disk (resumable with <code>/resume</code>)
<code>/compact</code>	Free up context by summarizing the conversation so far
<code>/context</code>	Show current context usage GATED
<code>/copy</code>	Copy Claude's last response to clipboard (or <code>/copy N</code> for the Nth-latest)
<code>/export</code>	Export the current conversation to a file or clipboard
<code>/focus</code>	Toggle focus view: just your prompt, summary, and response
<code>/fork</code>	Spawn a background agent that inherits the full conversation
<code>/goal</code>	Set a goal — keep working until the condition is met GATED
<code>/memory</code>	Edit CLAUDE.md files and memory settings
<code>/pause-memory</code>	Pause automemory for this session GATED
<code>/recap</code>	Generate a one-line session recap now
<code>/rename</code>	Rename the current conversation
<code>/resume</code>	Resume a previous conversation
<code>/rewind</code>	Restore the code and/or conversation to a previous point

Model & effort6	
<code>/advisor</code>	Let Claude consult a stronger model at key moments GATED
<code>/effort</code>	Set effort level for model usage
<code>/fast</code>	Toggle fast mode: the same model, answering faster. GATED
<code>/model</code>	Set the AI model for Claude Code GATED
<code>/plan</code>	Enable plan mode or view the current session plan
<code>/ultraplan</code>	Draft an editable plan in Claude Code on the web.

Code & review6	
<code>/autofix-pr</code>	Monitor and autofix any issues with the current PR GATED
<code>/batch</code>	Research and plan a large-scale change, then execute it in parallel across 5–30 isolated worktree agents that each open a PR. ADD-ON
<code>/diff</code>	View uncommitted changes and per-turn diffs
<code>/init</code>	Initialize a new CLAUDE.md file with codebase documentation ADD-ON
<code>/install-github-app</code>	Set up Claude GitHub Actions for a repository
<code>/ultrareview</code>	Run a cloud-hosted multi-agent code review of the current branch, or of a pull request, and print the findings. GATED

Background & parallel work16	
<code>/background</code>	Send this session to the background and free the terminal
<code>/cloud-plugins</code>	Choose whether cloud sessions use the plugins enabled on this machine
<code>/daemon</code>	Manage background services and routines
<code>/desktop</code>	Continue the current session in Claude Desktop GATED
<code>/list-agents</code>	List subagents, teammates, and other Claude sessions you can message
<code>/loops</code>	List, create, and delete loops
<code>/mobile</code>	Show QR code to download the Claude mobile app
<code>/remote-control</code>	Control this session from your phone or claude.ai/code GATED
<code>/remote-env</code>	Choose the default environment for cloud agents GATED
<code>/session</code>	Show cloud session URL and QR code GATED
<code>/stop</code>	Stop this background session; transcript and worktree are kept
<code>/subtask</code>	Send a subagent off with your full context; its result comes back here
<code>/tasks</code>	View and manage everything running in the background
<code>/teleport</code>	Send this session to the cloud, or resume one from claude.ai GATED
<code>/workflow-launch-exec</code>	Execute a server-launched workflow handoff (workflow_launch event sessions only) GATED
<code>/workflows</code>	Browse running and completed workflows

Skills, plugins & agents17	
<code>/agents</code>	(removed) Ask Claude to create/manage subagents, or edit .claude/agents/
<code>/artifacts</code>	Browse your published and shared artifacts
<code>/design</code>	Grant or revoke Claude agent access to your Design projects
<code>/design-consent</code>	Grant Claude agent access to your Design projects GATED
<code>/design-login</code>	Authorize design-system access for <code>/design-sync</code> with your claude.ai account
<code>/design-revoke</code>	Revoke Claude agent access to your Design projects GATED
<code>/design-sync</code>	Push your design system components to claude.ai/design ADD-ON
<code>/fewer-permission-prompts</code>	Scan your transcripts for common read-only Bash and MCP tool calls, then add a prioritized allowlist to project .claude/settings.json to reduce permission prompts. ADD-ON
<code>/insights</code>	Generate a report analyzing your Claude Code sessions ADD-ON
<code>/plugin</code>	Manage Claude Code plugins
<code>/plugin-types</code>	Write claude-code.d.ts and claude-code-mcp.d.ts: the plugin API's TypeScript declarations and the inputs of the connected MCP tools, for typing a hooks module against this session
<code>/powerup</code>	Discover Claude Code features through quick interactive lessons
<code>/reload-plugins</code>	Activate pending plugin changes in the current session
<code>/reload-skills</code>	Pick up skills added or changed on disk during this session
<code>/skill-doctor</code>	Show which loaded skills are unused and costing context GATED
<code>/skills</code>	List available skills
<code>/team-onboarding</code>	Help teammates ramp on Claude Code with a guide from your usage GATED ADD-ON

Integrations8	
<code>/chrome</code>	Open Claude in Chrome settings
<code>/ide</code>	Manage IDE integrations and show status
<code>/import</code>	Import config from another AI coding agent GATED
<code>/install-slack-app</code>	Install the Claude Slack app
<code>/mcp</code>	Manage MCP servers GATED
<code>/setup-bedrock</code>	Reconfigure Amazon Bedrock authentication, region, or model pins GATED
<code>/setup-vertex</code>	Reconfigure Google Vertex AI authentication, project, region, or model pins GATED
<code>/web-setup</code>	Set up Claude Code on the web with your GitHub account GATED

Configuration24	
<code>/auto-mode-setup</code>	Teach auto mode about your environment, plus optional rule tweaks GATED
<code>/color</code>	Set the prompt bar color for this session
<code>/config</code>	Set a setting by key GATED
<code>/debug</code>	Enable debug logging for this session and help diagnose issues ADD-ON
<code>/doctor</code>	Health-check your setup and fix issues: installation, unused extensions, duplicated or bloated memory files, slow hooks, updates, permissions
<code>/heapdump</code>	Dump the JS heap to ~/Desktop GATED
<code>/hooks</code>	View hook configurations for tool events
<code>/install</code>	Install Claude Code native build
<code>/keybindings</code>	Open your keyboard shortcuts file
<code>/permissions</code>	Manage allow and deny tool permission rules
<code>/privacy-settings</code>	View and update your privacy settings
<code>/radio</code>	Listen to Claude FM lo-fi radio
<code>/release-notes</code>	View release notes
<code>/sandbox</code>	Show and change sandboxing for this session: whether commands run sandboxed, and what is allowed without asking. GATED
<code>/scroll-speed</code>	Adjust mouse wheel scroll speed
<code>/statusline</code>	Set up Claude Code's status line UI ADD-ON
<code>/stickers</code>	Order Claude Code stickers
<code>/terminal-setup</code>	Set up the key binding for newlines in your terminal, and quiet the audible bell.
<code>/theme</code>	Change the theme
<code>/tui</code>	Set the terminal UI renderer (default fullscreen)
<code>/update</code>	Switch to the latest version (conversation continues) GATED
<code>/version</code>	Show this session's version (autoupdate may have a newer one)
<code>/voice</code>	Toggle voice mode GATED
<code>/wellbeing</code>	Configure optional break reminders and quiet-hours nudges

Account & usage14	
<code>/bug</code>	Report a bug or share your conversation
<code>/exit</code>	End the session.
<code>/extra-usage</code>	Renamed to <code>/usage-credits</code> GATED
<code>/feedback</code>	Send feedback to Anthropic or report a bug
<code>/help</code>	Show help and available commands
<code>/login</code>	Sign in with your Anthropic account
<code>/logout</code>	Sign out from your Anthropic account
<code>/passes</code>	Share a free week of Claude Code with friends and earn usage credits. GATED
<code>/pro-trial-expired</code>	Options shown when the Pro plan Claude Code trial has ended GATED
<code>/rate-limit-options</code>	Show options when rate limit is reached GATED
<code>/status</code>	Show Claude Code status including version, model, account, API connectivity, and tool statuses
<code>/upgrade</code>	Upgrade to Max for higher rate limits and more Opus
<code>/usage</code>	Show session cost, plan usage, and what's contributing to your limits GATED
<code>/usage-credits</code>	Configure usage credits or request them from your admin when you hit a limit GATED
<code>/cost</code>	<i>now an alias of /usage</i>
<code>/stats</code>	<i>now an alias of /usage</i>
<code>/extra-usage</code>	<i>renamed to /usage-credits</i>

Shortcuts & symbols14	
Esc	Stop what Claude is doing, without ending the session.
Esc Esc	Open the rewind menu.
Shift + Tab	Cycle permission modes: normal, auto-accept, plan.
Tab	Toggle extended thinking.
Ctrl + R	Search back through your command history.
Ctrl + V	Paste an image from the clipboard.
Ctrl + L	Clear the terminal without clearing the conversation.
Ctrl + C	Interrupt. Twice to exit.
Ctrl + D	Exit the session.
@filename	Reference a file or directory in your prompt.
#content	Add a line to CLAUDE.md without leaving the session.
!command	Run a shell command directly and keep its output in context.
"think harder"	Raise effort for a single turn.
"ultrathink"	Maximum reasoning depth for a single turn.

Starting Claude from the terminal.

Subcommands		18
claude agents [options]	Manage background agents	
claude attach <id>	Open a background session in this terminal. <id> is the short id that `claude --bg` prints and `claude agents` lists	
claude auth	Manage authentication	
claude auto-mode	Inspect or reset auto mode classifier configuration	
claude doctor	Check the health of your Claude Code installation. Reads settings files in the current directory without a trust prompt. For a full checkup that can also fix issues, run /doctor in a session.	
claude gateway [options]	Run the enterprise auth/telemetry gateway	
claude import [options] [source]	Import config from another AI coding agent into Claude Code	
claude install [options] [target]	Install Claude Code native build. Use [target] to specify version (stable, latest, or specific version)	
claude logs <id>	Print a background session's recent terminal output	
claude mcp	Configure and manage MCP servers	
claude plugin plugins	Manage Claude Code plugins	
claude project	Manage Claude Code project state	
claude respawn [options] [id]	Restart a background session, or all of them with --all, so it runs the current Claude Code version	
claude rm <id>	Delete a background session, and its worktree when that is safe. Works on sessions that have already exited	
claude setup-token	Set up a long-lived authentication token (requires Claude subscription)	
claude stop kill <id>	Stop a background session. Its conversation is kept: `claude attach <id>` opens it again, `claude --resume` works once it is stopped	
claude ultrareview [options] [target]	Run a cloud-hosted multi-agent code review of the current branch (or a PR number / base branch) and print the findings	
claude update upgrade	Check for updates and install if available	

Options	65
--add-dir <directories...>	Additional directories to allow tool access to
--agent <agent>	Agent for the current session. Overrides the 'agent' setting.
--agents <json>	JSON object defining custom agents (e.g. '{"reviewer": {"description": "Reviews code", "prompt": "You are a code reviewer"}}')
--allow-dangerously-skip-permissions	Enable bypassing all permission checks as an option, without it being enabled by default. Recommended only for sandboxes with no internet access.
--allowedTools, --allowed-tools <tools...>	Comma or space-separated list of tool names to allow (e.g. "Bash(git *) Edit")
--append-system-prompt <prompt>	Append a system prompt to the default system prompt
--autocompact <auto tokens>	Auto-compact window size (auto, or 100k–1M tokens)
--ax-screen-reader	Render screen-reader friendly output (flat text, no decorative borders or animations).
--bg, --background	Start the session in the background and return immediately. Prints the id that 'claude attach', 'logs', 'stop' and 'rm' take; 'claude agents' lists them. With --resume <session-id>, continues that session in the background under the same ID, or starts a copy and says so when the session is already running
--bare	Minimal mode: skip hooks, LSP, plugin sync, attribution, auto-memory, background prefetches, keychain reads, and CLAUDE.md auto-discovery. Sets CLAUDE_CODE_SIMPLE=1. Anthropic auth is strictly ANTHROPIC_API_KEY or apiKeyHelper via --settings (OAuth and keychain are never read). 3P providers (Bedrock/Vertex/Foundry) use their own credentials. Skills still resolve via /skill-name. Explicitly provide context via: --system-prompt[-file], --append-system-prompt[-file], --add-dir (CLAUDE.md dirs), --mcp-config, --settings, --agents, --plugin-dir.
--betas <betas...>	Beta headers to include in API requests (API key users only)
--brief	Enable SendUserMessage tool for agent-to-user communication
--chrome	Enable Claude in Chrome integration
--cloud [description session_id url]	Create a cloud session with the given description, or attach to an existing one by session ID or claude.ai/code URL
-c, --continue	Continue the most recent conversation in the current directory
--dangerously-skip-permissions	Bypass all permission checks. Recommended only for sandboxes with no internet access.
-d, --debug [filter]	Enable debug mode with optional category filtering (e.g., "api,hooks" or "!p,!file")
--debug-file <path>	Write debug logs to a specific file path (implicitly enables debug mode)
--disable-slash-commands	Disable all skills
--disallowedTools, --disallowed-tools <tools...>	Comma or space-separated list of tool names to deny (e.g. "Bash(git *) Edit")
--effort <level>	Effort level for the current session (low, medium, high, xhigh, max)
--environment <environment_id>	Create a new cloud session that runs on the given self-hosted environment (ccpool,...).
--exclude-dynamic-system-prompt-sections	Move per-machine sections (cwd, env info, memory paths, git status) from the system prompt into the first user message. Improves cross-user prompt-cache reuse. Only applies with the default system prompt (ignored with --system-prompt). (default: false)
--fallback-model <model>	Enable automatic fallback to specified model(s) when the default model is overloaded or not available. Accepts a comma-separated list to try each in order. Re-tries the primary at the start of each user turn. (only works with --print)
--file <specs...>	File resources to download at startup. Format: file_id:relative_path (e.g., --file file_abc:doc.txt file_def:img.png)
--fork-session	When resuming, create a new session ID instead of reusing the original (use with --resume or --continue)
--forward-subagent-text	Forward subagent text and thinking blocks as assistant/user messages with parent_tool_use_id set (only works with --print and --output-format=stream-json)
--from-pr [value]	Resume a session linked to a PR by PR number/URL, or open interactive picker with optional search term
-h, --help	Display help for command
--ide	Automatically connect to IDE on startup if exactly one valid IDE is available
--include-hook-events	Include all hook lifecycle events in the output stream (only works with --output-format=stream-json)
--include-partial-messages	Include partial message chunks as they arrive (only works with --print and --output-format=stream-json)
--input-format <format>	Input format (only works with --print): "text" (default), or "stream-json" (realtime streaming input) (choices: "text", "stream-json")
--json-schema <schema>	JSON Schema for structured output validation. Example: {"type":"object","properties":{"name":{"type":"string"}}, "required":["name"]}
--max-budget-usd <amount>	Maximum dollar amount to spend on API calls (only works with --print)
--mcp-config <configs...>	Load MCP servers from JSON files or strings (space-separated)
--model <model>	Model for the current session. Provide an alias for the latest model (e.g. 'fable', 'opus', or 'sonnet') or a model's full name (e.g. 'claude-fable-5').
-n, --name <name>	Set a display name for this session (shown in the prompt box, /resume picker, and terminal title)
--no-chrome	Disable Claude in Chrome integration
--no-session-persistence	Disable session persistence - sessions will not be saved to disk and cannot be resumed (only works with --print)
--output-format <format>	Output format (only works with --print): "text" (default), "json" (single result), or "stream-json" (realtime streaming) (choices: "text", "json", "stream-json")
--permission-mode <mode>	Permission mode to use for the session (choices: "acceptEdits", "auto", "bypassPermissions", "manual", "dontAsk", "plan")
--permission-prompts <target>	Who answers permission prompts with --print: "host" (the SDK host or --permission-prompt-tool) or "none" (nobody: anything that would prompt is denied automatically; the permission mode still decides everything else) (choices: "host", "none", default: "host")
--plugin-dir <path>	Load a plugin from a directory or .zip for this session only (repeatable: --plugin-dir A --plugin-dir B.zip) (default: [])
--plugin-url <url>	Fetch a plugin .zip from a URL for this session only (repeatable: --plugin-url A --plugin-url B) (default: [])
-p, --print	Print response and exit (useful for pipes). Note: The workspace trust dialog is skipped when Claude is run in non-interactive mode (via -p, or when stdout is not a TTY, e.g. piped or redirected output). Only use this in directories you trust. Settings files that fail validation are silently ignored in this mode (no error dialog is shown).
--prompt-suggestions [value]	Enable prompt suggestions. In print/SDK mode, emits a prompt_suggestion message after each turn with a predicted next user prompt (choices: "true", "false", "1", "0", "yes", "no", "on", "off", preset: "true")
--remote-control [name]	Start an interactive session with Remote Control enabled (optionally named)
--remote-control-session-name-prefix <prefix>	Prefix for auto-generated Remote Control session names (default: hostname)

--replay-user-messages	Re-emit user messages from stdin back on stdout for acknowledgment (only works with --input-format=stream-json and --output-format=stream-json)
--restricted	Restricted mode: removes the built-in tools that run commands or code (Bash, PowerShell, REPL and the other code-running tools) and WebFetch unless --tools names them, and ignores user, project and local settings files (managed settings and --settings still apply; add --strict-mcp-config to skip MCP servers too). Also confines the file tools to the working directories (--add-dir included), refuses bypassPermissions, and lets only a person or the configured permission handler approve writes to settings, git and tool-configuration files.
-r, --resume [value]	Resume a conversation by session ID, or open interactive picker with optional search term
--safe-mode	Start with all customizations (CLAUDE.md, skills, plugins, hooks, MCP servers, custom commands and agents, output styles, workflows, custom themes, keybindings, and more) disabled — useful for troubleshooting a broken configuration. Admin-managed (policy) settings still apply. Auth, model selection, built-in tools, and permissions work normally. Sets CLAUDE_CODE_SAFE_MODE=1.
--session-id <uuid>	Use a specific session ID for the conversation (must be a valid UUID)
--setting-sources <sources>	Comma-separated list of setting sources to load (user, project, local).
--settings <file-or-json>	Path to a settings JSON file or a JSON string to load additional settings from
--strict-mcp-config	Only use MCP servers from --mcp-config, ignoring all other MCP configurations
--system-prompt <prompt>	System prompt to use for the session
--system-prompt-snapshot <on off>	Record the system prompt once per conversation and reuse it verbatim on every request and resume (recommended: on). By default it is on for the built-in prompt, and passing --system-prompt or --append-system-prompt turns it off so the given text applies fresh each launch. on: an existing record in the conversation is sent as-is (a later launch's different --system-prompt/--append-system-prompt is ignored until compaction); otherwise the prompt is rendered with the append included, sent, and recorded. off: never record. No effect where system-prompt recording is not yet enabled. (choices: "on", "off")
--teleport [session]	Resume a teleport session, optionally specify session ID
--tmux	Create a tmux session for the worktree (requires --worktree). Uses iTerm2 native panes when available; use --tmux=classic for traditional tmux.
--tools <tools...>	Specify the list of available tools from the built-in set. Use "" to disable all tools, "default" to use all tools, or specify tool names (e.g. "Bash,Edit,Read").
--verbose	Override verbose mode setting from config
-v, --version	Output the version number
-w, --worktree [name]	Create a new git worktree for this session (optionally specify a name)

