



Slash Command Reference: Top 10 AI Companies

Table of Contents

About this reference.....	2
1. OpenAI.....	3
ChatGPT desktop app (task composer).....	3
Codex CLI.....	4
2. Anthropic.....	6
Claude Code.....	6
Claude.ai (web, desktop, and mobile chat).....	10
3. Google.....	11
Gemini CLI.....	11
Gemini app (consumer chat).....	12
4. Microsoft.....	13
GitHub Copilot CLI.....	13
GitHub Copilot Chat (by environment).....	15
Microsoft 365 Copilot (Word, Outlook, Teams).....	16
5. Meta.....	18
Meta AI (WhatsApp, Instagram, Facebook Messenger).....	18
6. xAI.....	19
Grok Build (coding CLI).....	19
Grok (consumer chat, on X and grok.com).....	20
7. Perplexity.....	21
8. Mistral AI.....	22
Mistral Vibe (coding CLI, part of Le Chat's Code surface).....	22
Le Chat (consumer chat).....	23
9. Amazon.....	24
Amazon Q Developer CLI.....	24
10. DeepSeek.....	25
Let us find the right AI for you.....	25
Sources.....	26

Slash Command Reference: Top 10 AI Companies

Compiled July 2026

About this reference

This document catalogs the “/” (slash) commands published by the ten best-known AI product companies, across both their consumer chat interfaces and their developer-facing coding agents/CLIs.

Two patterns emerged during research. Developer-facing coding agents (OpenAI Codex CLI, Anthropic Claude Code, Google Gemini CLI, GitHub Copilot CLI, xAI Grok Build, Mistral Vibe) all publish extensive, officially documented slash-command references, typically 20-90 commands each. Consumer chat products (ChatGPT web, Claude.ai, the Gemini app, Meta AI, Grok on X, Perplexity, Le Chat, Microsoft 365 Copilot, DeepSeek) rely much more heavily on natural-language prompts, @ mentions, and UI buttons; where slash commands exist at all in these products, they are often a small, loosely documented set, or are entirely community-invented rather than official.

Where a company’s official documentation did not provide an exhaustive list, this is noted explicitly rather than presented as complete.

.enAI

10.ChatGPT desktop app (task composer)

Source: OpenAI Developers / ChatGPT Learn, “Slash commands” reference (learn.chatgpt.com/codex/reference/slash-commands).

Command	Description
/approve	Approve one retry of a recent automatic-review denial, when automatic review is active.
/cloud	Run the task in the cloud, when cloud execution is available.
/cloud-environment	Choose the cloud environment for the task.
/compact	Compact the current task’s context.
/fast	Turn a catalog-provided Fast service tier on or off, when available.
/feedback	Open the feedback dialog to submit feedback and optionally include logs.
/fork	Copy a local task into a new local task or worktree.
/goal	Set a persistent goal for ChatGPT to work toward; use /plan first to shape it.
/ide-context	Turn shared IDE context on or off.
/init	Generate an AGENTS.md scaffold for the current project.
/local	Run the task in the selected local project.
/mcp	Open MCP status to view connected servers.
/memories	Configure whether the task can use or generate memories, when Memories is available.
/model	Choose the model for the current task.
/pet	Wake or tuck away the desktop pet.
/personality	Choose how ChatGPT responds, when the current model supports personalities.
/plan	Toggle plan mode for multi-step planning.
/project	Choose a project for new tasks.
/reasoning	Choose the reasoning effort for the current task.
/review	Start code review mode to review uncommitted changes or compare against a base branch.
/side	Start a temporary side conversation without interrupting the main task.
/status	Show the task ID, context usage, and rate limits.

/task	Start a task without a project.
/worktree	Run the task in a new Git worktree.

Enabled skills also appear in this same slash-command list, and custom saved prompts appear as /prompts:<name>.

Codex CLI

Source: OpenAI Developers, “Slash commands in Codex CLI” (developers.openai.com/codex/cli/slash-commands).

Command	Purpose
/permissions	Set what Codex can do without asking first.
/ide	Include open files, current selection, and other IDE context.
/keymap	Remap TUI keyboard shortcuts.
/vim	Toggle Vim mode for the composer.
/sandbox-add-read-dir	Grant sandbox read access to an extra directory (Windows only).
/agent	Switch the active agent thread.
/apps	Browse apps (connectors) and insert them into your prompt.
/plugins	Browse installed and discoverable plugins.
/hooks	View and manage lifecycle hooks.
/clear	Clear the terminal and start a fresh chat.
/archive	Archive the current session and exit Codex.
/delete	Permanently delete the current session and exit Codex.
/compact	Summarize the visible conversation to free tokens.
/copy	Copy the latest completed Codex output (also Ctrl+O).
/diff	Show the Git diff, including untracked files.
/exit	Exit the CLI (same as /quit).
/experimental	Toggle experimental features (e.g. subagents).
/approve	Approve one retry of a recent auto-review denial.
/memories	Configure memory use and generation.
/skills	Browse and use skills.
/import	Import Claude Code setup, project files, and recent chats.
/feedback	Send logs to the Codex maintainers.
/init	Generate an AGENTS.md scaffold in the current directory.
/logout	Sign out of Codex.
/mcp	List configured MCP tools.

/mention	Attach a file to the conversation.
/model	Choose the active model (and reasoning effort).
/fast	Toggle a Fast service tier when available.
/plan	Switch to plan mode and optionally send a prompt.
/goal	Set, pause, resume, view, or clear a task goal.
/personality	Choose a communication style (friendly, pragmatic, none).
/ps	Show experimental background terminals and their recent output.
/stop	Stop all background terminals.
/fork	Fork the current conversation into a new thread.
/side, /btw	Start an ephemeral side conversation.
/raw	Toggle raw scrollbar mode.
/resume	Resume a saved conversation from your session list.
/new	Start a new conversation inside the same CLI session.
/quit	Exit the CLI.
/review	Ask Codex to review your working tree.
/status	Display session configuration and token usage.
/usage	View account token usage or a rate-limit reset.
/debug-config	Print config layer and requirements diagnostics.
/statusline	Configure TUI status-line fields interactively.
/title	Configure terminal window/tab title fields.
/theme	Choose a syntax-highlighting theme.

2. Anthropic

Claude Code

Source: Anthropic, “Commands” reference (code.claude.com/docs/en/commands). Type `/` in any session to see the live, version-specific list.

Command	Description
<code>/add-dir <path></code>	Add a working directory for file access during the current session.
<code>/advisor [model\ off]</code>	Enable or disable the advisor tool, which consults a second model for guidance.
<code>/agents</code>	Prints a reminder to ask Claude to create or manage subagents.
<code>/autofix-pr [prompt]</code>	Spawn a Claude Code on the web session that watches the current branch’s PR and pushes fixes.
<code>/background [prompt]</code>	Detach the session to run as a background agent and free the terminal.
<code>/batch <instruction></code>	Orchestrate large-scale changes across a codebase in parallel.
<code>/branch [name]</code>	Create a branch of the current conversation at this point.
<code>/btw <question></code>	Ask a quick side question without adding to conversation history.
<code>/cd <path></code>	Move the session to a new working directory, preserving the prompt cache.
<code>/chrome</code>	Configure Claude in Chrome settings.
<code>/claude-api [migrate\ managed-agents-onboard]</code>	Manage Claude API / managed-agents migration.
<code>/clear [name]</code>	Start a new conversation with empty context.
<code>/code-review [low\ medium\ ...]</code>	Run a code review at a chosen depth.
<code>/color [color\ default]</code>	Set terminal color.
<code>/compact [instructions]</code>	Free up context by summarizing the conversation so far.
<code>/config [key=value ...]</code>	Open Settings to adjust theme, model, output style, etc.
<code>/context [all]</code>	Visualize current context usage as a colored grid.
<code>/copy [N]</code>	Copy the last (or Nth-latest) assistant response to clipboard.
<code>/cost</code>	Alias for <code>/usage</code> .

<code>/dataviz [request]</code>	Design guidance for charts, graphs, and dashboards.
<code>/debug [description]</code>	Enable debug logging and troubleshoot issues.
<code>/deep-research <question></code>	Fan out web searches, cross-check sources, synthesize a cited report.
<code>/design-login</code>	Authorize design-system access for <code>/design-sync</code> .
<code>/design-sync [hint]</code>	Convert your repo's React design system and upload it to Claude Design.
<code>/desktop</code>	Continue the current session in the Claude Code Desktop app.
<code>/diff</code>	Open an interactive diff viewer of uncommitted changes.
<code>/doctor</code>	Run a setup checkup that diagnoses and can fix issues.
<code>/effort [level\ auto]</code>	Set reasoning effort level.
<code>/exit</code>	Exit the CLI (alias <code>/quit</code>).
<code>/export [filename]</code>	Export the current conversation as plain text.
<code>/fast [on\ off]</code>	Toggle Fast mode.
<code>/feedback [report]</code>	Submit feedback or report a bug (aliases <code>/bug</code> , <code>/share</code>).
<code>/fewer-permission-prompts</code>	Scan transcripts for common read-only calls and add an allowlist.
<code>/focus</code>	Toggle a focused view of the last prompt and final response.
<code>/fork <directive></code>	Spawn a forked background subagent that inherits the conversation.
<code>/goal [condition\ clear]</code>	Set or clear a persistent goal.
<code>/heapdump</code>	Write a JS heap snapshot for diagnostics.
<code>/help</code>	Show help and available commands.
<code>/hooks</code>	View hook configurations for tool events.
<code>/ide</code>	Manage IDE integrations and show status.
<code>/init</code>	Initialize project with a <code>CLAUDE.md</code> guide.
<code>/insights</code>	Generate a report analyzing your Claude Code sessions.
<code>/install-github-app</code>	Install the Claude GitHub App for a repository.
<code>/install-slack-app</code>	Install the Claude Slack app.
<code>/keybindings</code>	Open your keyboard shortcuts file.

/login	Sign in to your Anthropic account.
/logout	Sign out from your Anthropic account.
/loop [interval] [prompt]	Run a prompt repeatedly while the session stays open.
/mcp [reconnect/enable/...]	Manage Model Context Protocol servers.
/memory	Edit CLAUDE.md memory files, manage auto-memory.
/mobile	Show a QR code to download the Claude mobile app.
/model [model]	Switch the AI model and save it as your default.
/passes	Share a free week of Claude Code with friends (if eligible).
/permissions	Manage allow, ask, and deny rules for tool permissions.
/plan [description]	Enter plan mode directly from the prompt.
/plugin [subcommand]	Manage Claude Code plugins.
/powerup	Discover features through interactive lessons.
/pr-comments [PR]	Removed in v2.1.91 — ask Claude to view PR comments directly instead.
/privacy-settings	View and update privacy settings (Pro/Max only).
/radio	Open Claude FM lo-fi radio in your browser.
/recap	Generate a one-line summary of the current session.
/release-notes	View the changelog in an interactive version picker.
/reload-plugins [--force]	Reload all active plugins without restarting.
/reload-skills	Re-scan skill and command directories for changes.
/remote-control	Make this session available for remote control from claude.ai (alias /rc).
/remote-env	Choose the default environment for cloud agents.
/rename [name]	Rename the current session.
/resume [session]	Resume a conversation by ID or name.
/review [PR]	Run a fast, read-only review of a GitHub pull request.
/rewind	Rewind the conversation and/or code to a previous point.

<code>/run</code>	Launch and drive your project's app to see a change working.
<code>/run-skill-generator</code>	Teach <code>/run</code> and <code>/verify</code> how to build and drive your app.
<code>/sandbox</code>	Toggle sandbox mode (supported platforms only).
<code>/schedule [description]</code>	Create, update, list, or run cloud-run routines.
<code>/scroll-speed</code>	Adjust mouse-wheel scroll speed.
<code>/security-review</code>	Analyze pending changes for security vulnerabilities.
<code>/setup-bedrock</code>	Configure Amazon Bedrock authentication and model pins.
<code>/setup-vertex</code>	Configure Google Cloud Vertex AI authentication and model pins.
<code>/simplify [target]</code>	Review changed code for cleanup opportunities and apply fixes.
<code>/skills</code>	List available skills.
<code>/stats</code>	Alias for <code>/usage</code> , opens on the Stats tab.
<code>/status</code>	Open Settings on the Status tab (version, model, account).
<code>/statusline</code>	Configure Claude Code's status line.
<code>/stickers</code>	Order Claude Code stickers.
<code>/stop</code>	Stop the current background session.
<code>/tasks</code>	View and manage everything running in the background (alias <code>/bashes</code>).
<code>/team-onboarding</code>	Generate a team onboarding guide from usage history.
<code>/teleport</code>	Pull a Claude Code on the web session into this terminal.
<code>/terminal-setup</code>	Configure terminal keybindings (Shift+Enter, etc).
<code>/theme</code>	Change the color theme.
<code>/tui [default\ fullscreen]</code>	Set the TUI display mode.
<code>/ultraplan <prompt></code>	Draft a plan in an ultraplan session for browser review.
<code>/ultrareview [PR]</code>	Run a deep, multi-agent code review in a cloud sandbox.
<code>/upgrade</code>	Open the upgrade page to switch plan tiers.

<code>/usage</code>	Show session cost, plan usage limits, and activity stats.
<code>/usage-credits</code>	Configure usage credits for after hitting a limit.
<code>/verify</code>	Confirm a code change works by building, running, and observing it.
<code>/vim</code>	Removed in v2.1.92 — use <code>/config</code> → Editor mode instead.
<code>/voice [hold\ tap\ ...]</code>	Configure voice input mode.
<code>/web-setup</code>	Connect GitHub to Claude Code on the web via local gh credentials.
<code>/workflows</code>	Open the workflow progress view.

Claude.ai (web, desktop, and mobile chat)

Anthropic has not published an extensive slash-command reference for the consumer Claude.ai product the way it has for Claude Code. The documented, native commands are minimal:

Command	Description
<code>/new</code>	Clear the current chat context and open a blank conversation.
<code>/help</code>	Surface a quick-reference panel of commands available in the current interface.

Most Claude.ai functionality (Projects, Artifacts, Skills, file uploads) is invoked through UI buttons and @ mentions rather than slash commands.

3. Google

Gemini CLI

Source: Google, “CLI commands” (geminicli.com/docs/reference/commands).

Command	Description
/about	Show version info.
/agents	Manage local and remote subagents (list, reload, enable, disable, config).
/auth	Change the authentication method.
/bug	File an issue about Gemini CLI.
/chat	Alias for /resume; browse sessions and manage checkpoints.
/clear	Clear the terminal screen and visible session history.
/commands	Manage custom slash commands loaded from .toml files.
/compress	Replace the entire chat context with a summary.
/copy	Copy the last CLI output to the clipboard.
/directory (/dir)	Manage workspace directories for multi-directory support.
/docs	Open Gemini CLI documentation in your browser.
/editor	Open a dialog for selecting supported editors.
/extensions	Manage extensions (config, disable, enable, install, list, etc).
/help (/?)	Display help information, including available commands.
/hooks	Manage lifecycle hooks.
/ide	Manage IDE integration (enable, disable, install, status).
/init	Analyze the current directory and generate a GEMINI.md context file.
/mcp	Manage configured MCP servers (auth, desc, list, reload, schema).
/memory	Manage hierarchical instructional memory from GEMINI.md files.
/model	Manage model configuration (manage, set).
/permissions	Manage folder trust settings and other permissions.
/plan	Switch to Plan Mode (read-only) and view the current plan.
/policies	Manage and list active policies.
/privacy	Display the Privacy Notice and set data-collection consent.
/quit (/exit)	Exit Gemini CLI (optionally ---lete to purge session history).
/restore	Restore project files to the state before a tool was executed.
/rewind	Navigate backward through conversation history (Esc twice).
/resume	Browse and resume previous conversation sessions.
/settings	Open the settings editor.

<code>/shells</code> (<code>/bashes</code>)	Toggle the background shells view.
<code>/setup-github</code>	Set up GitHub Actions to triage issues and review PRs with Gemini.
<code>/skills</code>	Manage Agent Skills (<code>disable</code> , <code>enable</code> , <code>list</code> , <code>reload</code>).
<code>/stats</code>	Display detailed session statistics (<code>session</code> , <code>model</code> , <code>tools</code>).
<code>/terminal-setup</code>	Configure terminal keybindings for multiline input.
<code>/theme</code>	Change the visual theme.
<code>/tools</code>	List available tools, optionally with descriptions.
<code>/upgrade</code>	Open the Gemini Code Assist upgrade page.
<code>/vim</code>	Toggle vim-style navigation and editing in the input area.

Gemini CLI also supports `@<path>` to inject file/directory content and `!<command>` to run shell commands directly.

Gemini app (consumer chat)

Google has not published an official, comprehensive slash-command list for the consumer Gemini app itself. Some Google surfaces that embed Gemini do use slash commands — for example, Firebase Studio’s chat panel supports shortcuts like `/generate` to trigger code-snippet generation, and developers building Google Chat apps can define their own custom slash commands (e.g., `/create`) — but these are feature-specific rather than a general Gemini-app command set.

4. Microsoft

GitHub Copilot CLI

Source: GitHub Docs, “GitHub Copilot CLI command reference.”

Command	Purpose
/add-dir PATH	Add a directory to the allowed list for file access.
/after, /every	Schedule a one-time or recurring prompt/command (experimental).
/agent	Browse and select from available custom agents.
/app	Launch the GitHub Copilot app, or show its download URL.
/ask QUESTION	Ask a quick side question without adding to history (experimental).
/allow-all (/yolo)	Enable all permissions (tools, paths, URLs).
/changelog (/release-notes)	Display the CLI changelog, optionally AI-summarized.
/chronicle	Session history tools and insights (standup, tips, improve, reindex).
/clear (/new, /reset)	Start a new conversation.
/clikit	Preview CLI business components (e.g. quota info).
/compact	Summarize conversation history to reduce context usage.
/context	Show context-window token usage and visualization.
/copy	Copy the last response to the clipboard.
/cwd (/cd)	Change or display the current working directory.
/delegate	Delegate changes to a remote repository via an AI-generated PR.
/diff	Review changes in the current directory.
/downgrade VERSION	Download and restart into a specific CLI version.
/env	Show loaded environment details (MCP servers, skills, hooks, etc).
/exit (/quit)	Exit the CLI.
/extensions (/extension)	Manage CLI extensions.
/experimental	Toggle, set, or show experimental features.
/feedback (/bug)	Provide feedback about the CLI.
/fleet	Enable parallel subagent execution of parts of a task.
/help	Show help for interactive commands.
/ide	Connect to an IDE workspace.

/init	Initialize Copilot custom instructions for the repository.
/instructions	View and toggle custom instruction files.
/keep-alive (/caffeinate)	Prevent the machine from sleeping during a session.
/list-dirs	Display allowed file-access directories.
/login / /logout	Log in to / out of Copilot.
/lsp	Manage the language-server configuration.
/mcp	Manage MCP server configuration.
/model (/models)	Select the AI model, or Auto.
/permissions	View or clear in-memory tool/path approvals.
/plan	Create an implementation plan before coding.
/plugin	Manage plugins and plugin marketplaces.
/pr	Manage pull requests for the current branch (view, create, fix, auto).
/remote	Enable or end remote steering of the session.
/rename	Rename the current session.
/research TOPIC	Run a deep research investigation using GitHub search and web sources.
/reset-allowed-tools	Reset the list of allowed tools.
/restart	Restart the CLI, preserving the current session.
/resume (/continue)	Switch to a different session.
/review	Run the code review agent to analyze changes.
/rubber-duck	Consult the rubber-duck agent for a second opinion.
/sandbox	Configure shell command sandboxing.
/search (/find)	Search the conversation timeline (experimental).
/security-review	Run the security review agent for vulnerabilities.
/session (/sessions)	Show session info and manage sessions.
/settings	Open settings, or set/reset a value inline.
/share (/export)	Share the session as Markdown, HTML, or a GitHub gist.
/skills	Manage skills for enhanced capabilities.
/statusline (/footer)	Configure status-line items.
/subagents (/agents)	Configure default and per-agent subagent models.
/tasks	View and manage tasks (subagents and shell commands).
/terminal-setup	Configure the terminal for multiline input.
/theme	View or set the color mode.
/tuikit	Preview TUIkit design-system components.

/undo (/rewind)	Rewind the last turn and revert file changes.
/update (/upgrade)	Update the CLI to the latest version.
/usage	Display session usage metrics per model.
/user	Manage the current GitHub user.
/version	Display version information.
/fork (/branch)	Fork the current session into a new one (experimental).
/worktree (/move)	Create a new Git worktree and switch to it (experimental).

GitHub Copilot Chat (by environment)

Source: GitHub Docs, “GitHub Copilot Chat cheat sheet.” Available commands vary by surface.

GitHub.com website:

Command	Description
/clear	Clear conversation.
/delete	Delete a conversation.
/new	Start a new conversation.
/rename	Rename a conversation.

Visual Studio Code:

Command	Description
/clear	Start a new chat session.
/explain	Explain how the code in your active editor works.
/fix	Propose a fix for problems in the selected code.
/fixTestFailure	Find and fix a failing test.
/help	Quick reference and basics of using GitHub Copilot.
/new	Create a new project.
/tests	Generate unit tests for the selected code.

Visual Studio:

Command	Description
/doc	Add a documentation comment for this symbol.
/explain	Explain how the code in your active editor works.
/fix	Propose a fix for problems in the selected code.
/help	Quick reference and basics of using GitHub Copilot.
/optimize	Analyze and improve running time of the selected code.

/tests	Generate unit tests for the selected code.
--------	--

JetBrains IDEs:

Command	Description
/chronicle	Review and analyze session history (standup, tips, search, improve).
/compact	Manually compress the current Copilot CLI session context.
/explain	Explain how the code in your active editor works.
/fix	Propose a fix for problems in the selected code.
/help	Quick reference and basics of using GitHub Copilot.
/remote	Enable/manage remote control of a Copilot CLI session (CLI sessions only).
/tests	Generate unit tests for the selected code.

Xcode:

Command	Description
/doc	Generate documentation for this symbol.
/explain	Provide an explanation for the selected code.
/fix	Suggest fixes for code errors and typos.
/simplify	Simplify the current code selection.
/tests	Create a unit test for the current code selection.

Microsoft 365 Copilot (Word, Outlook, Teams)

Microsoft has not published a single official, exhaustive list of the roughly 30 forward-slash prompts available across Microsoft 365 Copilot; the set also varies by app and license. The most consistently documented ones are:

Command	Description
/summarize	Summarize the current content (email thread, document, or channel conversation).
/draft	Enter drafting mode with content suggestions.
/rewrite	Rewrite selected text (Word, Outlook).
/catch up	Summarize what you missed in a Teams channel or meeting.
/translate	Translate the current content.
/format	Reformat the current content.
/schedule	Propose a meeting time or schedule an item.
/poll	Create a poll (Teams).
/suggest	Suggest edits or next steps.

Slash commands in Microsoft 365 Copilot require a paid Copilot add-on license and bypass the conversational interface, executing directly rather than as a prompt.

5. Meta

Meta AI (WhatsApp, Instagram, Facebook Messenger)

Meta has not published a formal slash-command reference. The commands that are documented in Meta's own help content are limited to:

Command	Description
/imagine <prompt>	Generate an image from a text prompt (e.g. in Instagram DMs or a WhatsApp chat).
/ai-options	Show the available commands/options in the current Meta AI surface.

Most Meta AI interaction is through @ mentions (e.g. @Meta AI in a WhatsApp group) or natural-language prompts rather than slash commands.

6. xAI

Grok Build (coding CLI)

Source: xAI Docs, “Modes and Commands” (docs.x.ai/build/modes-and-commands).
Distinct from the consumer Grok chat assistant.

Core TUI commands:

Command	What it does
<code>/quit (/exit)</code>	Quit the application.
<code>/home</code>	Return to the welcome screen.
<code>/new</code>	Start a new session.
<code>/resume</code>	Resume a previous session.
<code>/sessions</code>	Browse and pick from past sessions.
<code>/fork</code>	Fork the current session into a new one.
<code>/rename <title></code>	Rename the current session.
<code>/share</code>	Share the current session via URL.
<code>/session-info</code>	Show session info.
<code>/context</code>	View context usage.
<code>/model <name></code>	Switch the active model.
<code>/always-approve</code>	Toggle always-approve mode.
<code>/multiline</code>	Toggle multiline input.
<code>/compact [context]</code>	Compact conversation history.
<code>/compact-mode</code>	Toggle a denser UI layout.
<code>/theme [name]</code>	Switch the color theme.
<code>/feedback [text]</code>	Send feedback about the current session.
<code>/plan</code>	View the current session plan.
<code>/btw <question></code>	Ask a side question without interrupting.
<code>/rewind</code>	Rewind to an earlier point in the conversation.
<code>/usage</code>	Show token and credit usage for the session.
<code>/logout</code>	Sign out of the current account.
<code>/hooks, /plugins, /skills, /mcps</code>	Open the unified extensions modal, pre-selecting that tab.

Shell-provided commands:

Command	What it does
<code>/flush</code>	Flush conversation memory to disk now.

/memory	Search and edit persistent memory entries.
/dream	Trigger an offline memory-consolidation pass.
/imagine <prompt>	Generate an image from text.
/imagine-video <prompt>	Generate a video from text.

Any user-invokable skill also appears as a slash command (/<skill-name>).

Grok (consumer chat, on X and grok.com)

xAI has not published an official slash-command reference for the consumer Grok assistant. Grok's consumer surface is prompt- and button-driven rather than slash-command-driven.

7. Perplexity

Perplexity does not publish a single official, exhaustive slash-command reference. Reporting on the feature (and Perplexity's own in-product autocomplete) describes:

Command	Description
<code>/gpt</code>	Switch the underlying model to GPT for the current query (Pro/Max).
<code>/claude</code>	Switch the underlying model to Claude for the current query (Pro/Max).
<code>/summarize</code>	Generate a concise summary of a long document or page.

Beyond these, Pro and Max subscribers can define their own custom slash commands (name + instructions), which then behave like built-in ones and appear in the same autocomplete menu. This feature is reported as unavailable on the iOS app.

8. Mistral AI

Mistral Vibe (coding CLI, part of Le Chat's Code surface)

Source: Mistral Docs, "Commands and shortcuts"
(docs.mistral.ai/vibe/code/cli/commands-shortcuts).

Command	Alias	Description
/help		Show help for the current CLI session.
/config		Edit config settings.
/model		Select the active model.
/thinking		Select the thinking level.
/reload		Reload configuration, agent instructions, and skills from disk.
/clear		Clear conversation history.
/copy		Copy the last agent message to the clipboard.
/log		Show the path to the current interaction log file.
/debug		Toggle the debug console.
/compact		Compact conversation history by summarizing it.
/exit		Exit the application.
/status		Display agent statistics.
/teleport		Teleport the session to Vibe Code Web.
/proxy-setup		Configure proxy and SSL certificate settings.
/resume	/continue	Browse and resume past sessions.
/rename		Rename the current session.
/mcp	/connectors	Display available MCP servers/connectors and their tools.
/voice		Configure voice settings.
/leaninstall		Install the Lean 4 proof-assistant agent.
/unleaninstall		Uninstall the Lean 4 agent.
/rewind		Rewind to a previous message.

/loop		Schedule a recurring prompt.
/data-retention		Show data retention information.

Any local skill can also expose itself as a slash command.

Le Chat (consumer chat)

Mistral has not published an official slash-command reference for the consumer Le Chat interface itself (as distinct from its Vibe coding agent).

9. Amazon

Amazon Q Developer CLI

Source: AWS Documentation, “Chat commands” (docs.aws.amazon.com/amazonq/latest/qdeveloper/qq-chat-commands.html) and related AWS guidance. AWS’s page did not return a full itemized table during research; the commands below are those consistently documented across AWS’s own guidance.

Command	Description
<code>/clear</code>	Clear the current conversation.
<code>/compact</code>	Summarize conversation history to reduce context size.
<code>/context add</code>	Temporarily add files to the current chat session’s context.
<code>/context show --- pand</code>	Show current context, including nested/imported context.
<code>/context clear</code>	Clear all session context.
<code>/editor</code>	Open your default editor to compose a multi-line message.
<code>/help</code>	Show available commands.
<code>/hooks</code>	Manage context hooks.
<code>/load</code>	Restore a previously saved conversation state.
<code>/save</code>	Save the current conversation state.
<code>/model</code>	Choose the model used by the agent.
<code>/profile</code>	Switch between configured agent profiles.
<code>/tools</code>	Manage permissions for individual tools.
<code>/usage</code>	Show an estimate of context-window usage.
<code>/mcp</code>	Manage configured MCP servers.
<code>/reply</code>	Open your editor with the most recent assistant message quoted, for reply.
<code>/quit</code>	Exit the chat session.

10. DeepSeek

DeepSeek has not published an official slash-command system for its consumer chat product (chat.deepseek.com); the web interface exposes DeepThink and Search as toggle buttons rather than slash commands. Third-party, unofficial DeepSeek CLI wrappers (e.g., community projects on GitHub) define their own small command sets, typically:

Command	Description
/help	Show a help message.
/models	List available models.
/model <name>	Switch the active model.

These are conventions established by individual open-source projects, not by DeepSeek itself.

Let us find the right AI for you!

This guide should get you started and give you some advice about the basic AI companies and services, but this is only the beginning of the AI journey and it is changing fast. If you are ready to transform your personal capabilities or your company with focused AI tools that can help you become more effective and efficient then we are here to help.

Contact us today through email or on our website to schedule your free 30-Minute Introduction Call to see what AI can do for you!

Email: Sales@IntelliTechDynamics.com

Website: www.IntelliTechDynamics.com



Sources

- OpenAI Developers — Slash commands in Codex CLI (developers.openai.com/codex/cli/slash-commands)
- ChatGPT Learn — Slash commands (learn.chatgpt.com/codex/reference/slash-commands), Commands (learn.chatgpt.com/docs/reference/commands)
- Anthropic — Commands reference (code.claude.com/docs/en/commands)
- Google — Gemini CLI command reference (geminikli.com/docs/reference/commands/)
- GitHub Docs — Copilot CLI command reference (docs.github.com/en/copilot/reference/copilot-cli-reference/cli-command-reference), Copilot Chat cheat sheet (docs.github.com/en/copilot/reference/chat-cheat-sheet)
- xAI Docs — Modes and Commands (docs.x.ai/build/modes-and-commands)
- Mistral Docs — Commands and shortcuts (docs.mistral.ai/vibe/code/cli/commands-shortcuts)
- AWS Documentation — Amazon Q Developer chat commands (docs.aws.amazon.com/amazonq/latest/qdeveloper-ug/command-line-chat-commands.html)
- WhatsApp Help Center — About Meta AI (faq.whatsapp.com/2257017191175152)
- Microsoft Support — Microsoft 365 Copilot help (support.microsoft.com/en-us/microsoft-365-copilot/)