

APPENDIX C

settings.json Key Reference

Keys by family. Scope and defaults are in the settings reference; Chapter 5 covers precedence and Chapter 26 the managed-only keys.

Generated from the official docs snapshot of 2026-09-01 (Claude Code v2.1.257) by tools/gen_appendices.py. Do not edit by hand; rerun the script for the next edition.

- File locations: User = ~/.claude/settings.json; Project = .claude/settings.json; Local = .claude/settings.local.json
- File locations: Managed = managed-settings.json, MDM, or server-managed; Global config = ~/.claude.json
- Precedence (highest first): managed settings, --settings command line, project local, shared project, user

Model and responses

- `advisorModel`: model answering the advisor tool
- `alwaysThinkingEnabled`: false turns extended thinking off
- `availableModels`: allowlist of selectable models
- `effortLevel`: default effort level for unsaved models
- `enforceAvailableModels`: make Default obey availableModels

- **fallbackModel**: ordered backup model chain
- **fastMode**: turn fast mode on
- **fastModePerSessionOptIn**: require /fast each session
- **language**: response language instruction
- **model**: model new sessions start with
- **modelOverrides**: map model IDs to provider IDs
- **modelPicker**: custom /model picker rows
- **modelPicker.options**: rows with model, label, description
- **modelPicker.replaceBuiltInOptions**: show only custom rows
- **modelPricing**: report spend at contracted rates
- **modelPricing.multiplier**: scale every cost figure
- **modelPricing.overrides**: per-model input, output, cacheRead, cacheWrite rates
- **modelSettings**: per-model saved effortLevel
- **outputStyle**: selected output style name
- **promptCacheTtl**: main conversation cache lifetime 5m/1h
- **showThinkingSummaries**: show full thinking summaries
- **subagentPromptCacheTtl**: subagent and helper cache lifetime
- **switchModelsOnFlag**: auto-switch or pause on classifier flag
- **ultracode**: start sessions with ultracode on

Permission settings

- **allowManagedPermissionRulesOnly**: managed rules are the only rules
- **autoMode**: custom auto mode classifier rules
- **autoMode.environment**: classifier environment prose rules
- **autoMode.allow**: classifier allow prose rules
- **autoMode.soft_deny**: classifier soft deny prose rules
- **autoMode.hard_deny**: classifier hard deny prose rules
- **autoMode.classifyAllShell**: classify every shell command
- **disableAutoMode**: remove auto mode from cycle
- **permissions**: permission rules and starting mode object

- `permissions.allow`: tool uses approved without prompt
- `permissions.ask`: tool uses that always prompt
- `permissions.deny`: blocked tool uses
- `permissions.additionalDirectories`: extra working directories
- `permissions.defaultMode`: starting permission mode
- `permissions.disableBypassPermissionsMode`: block `bypassPermissions` mode
- `permissions.disableAutoMode`: alias of `disableAutoMode`
- `useAutoModeDuringPlan`: classifier reviews plan-mode shell commands
- `skipAutoPermissionPrompt`: skip auto mode one-time notice
- `skipDangerousModePermissionPrompt`: skip bypass confirmation dialog

Sandbox settings

- `sandbox`: Bash command isolation object
- `sandbox.enabled`: turn sandboxing on
- `sandbox.failIfUnavailable`: exit when sandbox can't start
- `sandbox.autoAllowBashIfSandboxed`: no prompt for sandboxed commands
- `sandbox.excludedCommands`: commands always run unsandboxed
- `sandbox.allowUnsandboxedCommands`: allow unsandboxed retry
- `sandbox.filesystem`: filesystem read/write path lists
- `sandbox.filesystem.allowWrite`: extra writable paths
- `sandbox.filesystem.denyWrite`: blocked write paths
- `sandbox.filesystem.denyRead`: blocked read paths
- `sandbox.filesystem.allowRead`: re-opened read paths
- `sandbox.filesystem.allowManagedReadPathsOnly`: honor only managed `allowRead`
- `sandbox.filesystem.disabled`: skip filesystem isolation
- `sandbox.ignoreViolations`: silence expected violation reports
- `sandbox.enableWeakerNestedSandbox`: run inside unprivileged container
- `sandbox.enableWeakerNetworkIsolation`: allow macOS trust service access
- `sandbox.allowAppleEvents`: allow Apple Events on macOS
- `sandbox.ripgrep`: custom ripgrep binary

- `sandbox.ripgrep.command`: path to ripgrep binary
- `sandbox.ripgrep.args`: arguments prepended to ripgrep
- `sandbox.bwrapPath`: bubblewrap binary path
- `sandbox.socatPath`: socat binary path
- `sandbox.credentials`: credential protection object
- `sandbox.credentials.files`: credential files to deny or mask
- `sandbox.credentials.files[].path`: credential file path
- `sandbox.credentials.files[].mode`: deny or mask
- `sandbox.credentials.files[].extract`: regex capturing value to mask
- `sandbox.credentials.files[].onExtractNoMatch`: warn, deny, or error
- `sandbox.credentials.files[].decode`: jwt token decoding
- `sandbox.credentials.files[].maskClaims`: JWT claims to mask
- `sandbox.credentials.files[].maskDuplicates`: mask verbatim copies
- `sandbox.credentials.files[].injectHosts`: hosts receiving real value
- `sandbox.credentials.envVars`: environment variables to deny or mask
- `sandbox.credentials.envVars[].name`: variable name
- `sandbox.credentials.envVars[].mode`: deny or mask
- `sandbox.credentials.envVars[].extract`: regex capturing value to mask
- `sandbox.credentials.envVars[].onExtractNoMatch`: warn, deny, or error
- `sandbox.credentials.envVars[].decode`: jwt token decoding
- `sandbox.credentials.envVars[].maskClaims`: JWT claims to mask
- `sandbox.credentials.envVars[].injectHosts`: hosts receiving real value
- `sandbox.credentials.allowPlaintextInject`: mask substitution over plain HTTP
- `sandbox.credentials.awsPairs`: group custom AWS credential variables
- `sandbox.credentials.awsPairs[].accessKeyIdVar`: access key variable name
- `sandbox.credentials.awsPairs[].secretAccessKeyVar`: secret key variable name
- `sandbox.credentials.awsPairs[].sessionTokenVar`: session token variable name
- `sandbox.credentials.sigv4`: unsupported AWS request handling
- `sandbox.credentials.sigv4.streaming`: streaming uploads deny/passthrough
- `sandbox.credentials.sigv4.presigned`: presigned URLs deny/passthrough
- `sandbox.credentials.sigv4.sigv4a`: SigV4A signatures deny/passthrough

- `sandbox.network`: network isolation object
- `sandbox.network.allowUnixSockets`: allowed macOS Unix socket paths
- `sandbox.network.allowAllUnixSockets`: allow every Unix socket
- `sandbox.network.allowLocalBinding`: bind localhost ports on macOS
- `sandbox.network.allowMachLookup`: allowed XPC/Mach service names
- `sandbox.network.allowedDomains`: pre-allowed outbound domains
- `sandbox.network.deniedDomains`: blocked outbound domains
- `sandbox.network.strictAllowlist`: deny instead of prompting
- `sandbox.network.allowManagedDomainsOnly`: lock allowlist to managed
- `sandbox.network.httpProxyPort`: custom HTTP proxy port
- `sandbox.network.socksProxyPort`: custom SOCKS proxy port
- `sandbox.network.tlsTerminate`: proxy terminates TLS
- `sandbox.network.tlsTerminate.caCertPath`: CA certificate path
- `sandbox.network.tlsTerminate.caKeyPath`: CA key path

Memory and context

- `autoCompactEnabled`: automatic compaction on/off
- `autoCompactWindow`: context tokens before compaction
- `autoMemoryDirectory`: auto memory storage directory
- `autoMemoryEnabled`: auto memory on/off
- `claudeMd`: managed inline CLAUDE.md text
- `claudeMdExcludes`: CLAUDE.md files to skip
- `env`: environment variables for sessions
- `fileCheckpointingEnabled`: file snapshots for /rewind
- `plansDirectory`: plan file directory
- `skillListingBudgetFraction`: context share for skill listing
- `skillListingMaxDescChars`: per-skill description cap

Interface and terminal

- `askUserQuestionTimeout`: question auto-continue timeout
- `autoContinueAtUsageLimit`: auto-continue after usage limit reset
- `autoScrollEnabled`: follow output in fullscreen
- `axScreenReader`: screen-reader friendly output
- `companyAnnouncements`: startup announcements
- `defaultShell`: shell for `!` commands
- `dialogExpiry`: forwarded dialog deadline
- `editorMode`: normal or vim key bindings
- `emojiCompletionEnabled`: `:shortcode`: emoji suggestions
- `fileSuggestion`: custom `@` autocomplete command
- `fileSuggestion.type`: always command
- `fileSuggestion.command`: shell command to run
- `footerLinksRegexes`: regex-driven footer link badges
- `footerLinksRegexes[].type`: always regex
- `footerLinksRegexes[].pattern`: regex with named captures
- `footerLinksRegexes[].url`: URL template
- `footerLinksRegexes[].label`: badge label template
- `keybindingFlavor`: `Ctrl+W` behavior classic/readline
- `prefersReducedMotion`: reduce interface animations
- `promptSuggestionEnabled`: show prompt suggestions
- `respectGitignore`: `@ picker` respects `.gitignore`
- `respondToBashCommands`: Claude replies to `!` output
- `showClearContextOnPlanAccept`: clear-context plan approval option
- `showTurnDuration`: show turn duration message
- `spellcheck`: prompt input spell checking
- `spellcheck.enabled`: turn spell checking on
- `spellcheck.checker`: `aspell`, `hunspell`, `ispell`, `auto`
- `spellcheck.language`: dictionary name
- `spellcheck.color`: underline color

- `spinnerTipsEnabled`: show spinner tips
- `spinnerTipsOVERRIDE`: custom spinner tips
- `spinnerTipsOVERRIDE.tips`: tip strings or objects
- `spinnerTipsOVERRIDE.tips[].id`: stable tip identifier
- `spinnerTipsOVERRIDE.tips[].text`: tip text
- `spinnerTipsOVERRIDE.tips[].cooldownSessions`: sessions between showings
- `spinnerTipsOVERRIDE.tips[].priority`: tie-break priority
- `spinnerTipsOVERRIDE.tipsFile`: JSON file of tips
- `spinnerTipsOVERRIDE.label`: tip prefix label
- `spinnerTipsOVERRIDE.excludeDefault`: hide built-in tips
- `spinnerVerbs`: custom spinner verbs
- `spinnerVerbs.verbs`: verb strings
- `spinnerVerbs.mode`: append or replace
- `statusLine`: custom status line command
- `statusLine.type`: always command
- `statusLine.command`: status line command
- `statusLine.padding`: horizontal spacing characters
- `statusLine.refreshInterval`: periodic re-run seconds
- `statusLine.hideVimModeIndicator`: hide built-in vim indicator
- `subagentStatusLine`: custom subagent row command
- `subagentStatusLine.type`: always command
- `subagentStatusLine.command`: row-rewriting command
- `syntaxHighlightingDisabled`: plain-text diffs and code
- `terminalProgressBarEnabled`: terminal progress indicator
- `terminalTitleFromRename`: tab title follows session name
- `theme`: interface color theme
- `tui`: default or fullscreen renderer
- `verbose`: full tool output
- `viewMode`: default, verbose, or focus view
- `vimInsertModeRemaps`: INSERT-mode sequences to Escape
- `voice`: voice dictation object

- `voice.enabled`: dictation on
- `voice.autoSubmit`: submit on release in hold mode
- `voice.mode`: hold or tap
- `voiceEnabled`: deprecated dictation Boolean
- `wheelScrollAccelerationEnabled`: mouse-wheel acceleration

Git and attribution

- `attribution`: commit and PR attribution object
- `attribution.commit`: commit trailer text
- `attribution.pr`: PR description attribution text
- `attribution.sessionUrl`: include claude.ai session link
- `includeCoAuthoredBy`: deprecated attribution toggle
- `includeGitInstructions`: built-in git instructions and snapshot
- `prUrlTemplate`: PR link URL template

Hooks and automation

- `allowedHttpHookUrls`: HTTP hook URL allowlist
- `allowManagedHooksOnly`: only managed hooks run
- `disableAllHooks`: turn all hooks off
- `disableWorkflows`: dynamic workflows off for everyone
- `enableWorkflows`: personal dynamic workflows toggle
- `hooks`: lifecycle hook definitions by event
- `hooks.<Event>[].matcher`: hook matcher pattern
- `hooks.<Event>[].hooks`: handler list
- `hooks.<Event>[].hooks[].type`: command, prompt, agent, http, mcp_tool
- `httpHookAllowedEnvVars`: env vars HTTP hooks may use
- `workflowKeywordTriggerEnabled`: ultracode keyword triggers workflow
- `workflowSizeGuideline`: target agent count for workflows

Plugins and skills

- `disableBundledSkills`: remove bundled skills and workflows
- `disableSkillShellExecution`: block inline shell in skills
- `skillOverrides`: per-skill visibility override
- `syncClaudeAiSkills`: sync claude.ai skills
- `allowedChannelPlugins`: channel plugin allowlist
- `allowedChannelPlugins[].marketplace`: marketplace name
- `allowedChannelPlugins[].plugin`: plugin name
- `blockedMarketplaces`: blocked marketplace sources
- `channelsEnabled`: allow channels for organization
- `disableCommandPluginSources`: block command-sourced plugins
- `pluginSuggestionMarketplaces`: marketplaces allowed to suggest plugins
- `pluginTrustMessage`: custom plugin trust warning text
- `strictKnownMarketplaces`: marketplace source allowlist
- `allowedMarketplaces`: alias of `strictKnownMarketplaces`
- `strictPluginOnlyCustomization`: lock customization to plugins/managed
- `strictPluginOnlyCustomization.skills`: lock skills surface
- `strictPluginOnlyCustomization.agents`: lock agents surface
- `strictPluginOnlyCustomization.hooks`: lock hooks surface
- `strictPluginOnlyCustomization.mcp`: lock MCP surface
- `enabledPlugins`: per-plugin on/off map
- `extraKnownMarketplaces`: register marketplaces by name
- `additionalMarketplaces`: alias of `extraKnownMarketplaces`
- `extraKnownMarketplaces.<name>.source`: marketplace source object
- `extraKnownMarketplaces.<name>.source.source`: github, git, url, file, directory, settings
- `extraKnownMarketplaces.<name>.source.repo`: GitHub repository
- `extraKnownMarketplaces.<name>.source.url`: git or marketplace URL
- `extraKnownMarketplaces.<name>.source.path`: local file or directory path
- `extraKnownMarketplaces.<name>.source.headers`: HTTP headers for URL source

- `extraKnownMarketplaces.<name>.source.headersHelper`: command producing headers
- `extraKnownMarketplaces.<name>.source.skipLfs`: skip Git LFS downloads
- `extraKnownMarketplaces.<name>.source.name`: inline marketplace name
- `extraKnownMarketplaces.<name>.source.plugins`: inline plugin entries
- `extraKnownMarketplaces.<name>.autoUpdate`: background marketplace refresh
- `pluginConfigs`: stored plugin configuration answers
- `pluginConfigs.<id>.options`: plugin option values
- `pluginConfigs.<id>.mcpServers`: per-server user configuration

MCP

- `allowAllClaudeAiMcps`: load connectors alongside `managed-mcp.json`
- `allowedMcpServers`: MCP server allowlist
- `allowedMcpServers[].serverName`: match by server name
- `allowedMcpServers[].serverCommand`: match by exact command
- `allowedMcpServers[].serverUrl`: match by URL pattern
- `allowManagedMcpServersOnly`: managed allowlist only
- `deniedMcpServers`: MCP server denylist
- `deniedMcpServers[].serverName`: match by server name
- `deniedMcpServers[].serverCommand`: match by exact command
- `deniedMcpServers[].serverUrl`: match by URL pattern
- `disableClaudeAiConnectors`: stop fetching `claude.ai` connectors
- `disabledMcpjsonServers`: rejected `.mcp.json` servers
- `enableAllProjectMcpServers`: approve all `.mcp.json` servers
- `enabledMcpjsonServers`: approved `.mcp.json` servers

Agents, sessions, and worktrees

- `agent`: run main thread as named subagent
- `crossSessionInbound`: accept, hold, or refuse messages
- `disableAgentView`: turn off background agents

- `isolatePeerMachines`: approve cross-machine messages
- `processWrapper`: corporate launcher prefix
- `teammateMode`: teammate display mode
- `worktree`: git worktree configuration
- `worktree.baseRef`: fresh or head branch base
- `worktree.symLinkDirectories`: directories symlinked into worktrees
- `worktree.sparsePaths`: sparse-checkout directories
- `worktree.bgIsolation`: background edit isolation `worktree/none`

Remote, desktop, and notifications

- `agentPushNotifEnabled`: push when Claude decides
- `awaySummaryEnabled`: session recap on return
- `disableArtifact`: deprecated Artifact tool off
- `disableDeepLinkRegistration`: skip `claude-cli://` registration
- `disableDesktopLocalSessions`: no on-device desktop sessions
- `disableRemoteControl`: turn Remote Control off
- `enableArtifact`: false turns Artifact tool off
- `inputNeededNotifEnabled`: push when input needed
- `preferredNotifChannel`: bell or desktop notification channel
- `remote.defaultEnvironmentId`: default cloud environment
- `remoteControlAtStartup`: auto-connect Remote Control
- `sshConfigs`: Desktop SSH connections
- `sshConfigs[].id`: connection identifier
- `sshConfigs[].name`: display name
- `sshConfigs[].sshHost`: SSH host
- `sshConfigs[].sshPort`: SSH port
- `sshConfigs[].sshIdentityFile`: identity file path
- `sshHostAllowList`: allowed Desktop SSH hosts

Authentication and providers

- `apiKeyHelper`: command producing API credential
- `awsAuthRefresh`: command refreshing .aws credentials
- `awsCredentialExport`: command printing AWS credentials JSON
- `forceLoginMethod`: claudeai, console, or gateway
- `forceLoginGatewayUrl`: cloud gateway login URL
- `forceLoginOrgUUID`: required claude.ai organization UUID(s)
- `gcpAuthRefresh`: command refreshing Google Cloud credentials
- `otelHeadersHelper`: command generating OpenTelemetry headers

Updates and versioning

- `autoUpdatesChannel`: latest or stable channel
- `minimumVersion`: update floor version
- `requiredMaximumVersion`: refuse to start above version
- `requiredMinimumVersion`: refuse to start below version

Tools

- `browserExternalPageTools`: keep tools off external pages
- `disableBrowserExternalNavigation`: Browser pane localhost only
- `disableMobileSimulatorTools`: block iOS Simulator tools

Privacy and telemetry

- `cleanupPeriodDays`: transcript retention days
- `desktopSessionCleanupPeriodDays`: desktop/Cowork transcript age limit
- `feedbackDrafts`: notify, quiet, or off
- `feedbackSurveyRate`: session survey probability
- `skipWebFetchPreflight`: skip WebFetch hostname check

Enterprise and managed settings

- `disableSideloadFlags`: reject sideload CLI flags
- `forceRemoteSettingsRefresh`: block startup until fetch
- `managedSourcesBehavior`: first-wins or merge sources
- `parentSettingsBehavior`: handle host-supplied settings
- `policyHelper`: executable computing managed settings
- `policyHelper.path`: helper executable path
- `policyHelper.timeoutMs`: helper timeout milliseconds
- `policyHelper.refreshIntervalMs`: background re-run interval
- `wslInheritsWindowsSettings`: WSL reads Windows policy chain

Global config settings (~/.claude.json)

- `autoConnectIde`: auto-connect IDE from external terminal
- `autoInstallIdeExtension`: auto-install IDE extension
- `diffTool`: auto or terminal diff display
- `externalEditorContext`: last response in Ctrl+G editor
- `permissionExplainerEnabled`: Ctrl+E command explanation
- `teammateDefaultModel`: removed in v2.1.234