Configuration Reference

Use this page as a searchable reference for Codex configuration files. For conceptual guidance and examples, start with Basic Config and Advanced Config.

`config.toml`

User-level configuration lives in `~/.codex/config.toml`.

<ConfigTable options={[{ key: “model”, type: “string”, description: “Model to use (e.g., `gpt-5-codex`).”, }, { key: “review_model”, type: “string”, description: “Model override used by `/review`.”, }, { key: “model_provider”, type: “string”, description: “Provider id from `model_providers` (default: `openai`).”, }, { key: “model_context_window”, type: “number”, description: “Context window tokens available to the active model.”, }, { key: “model_auto_compact_token_limit”, type: “number”, description: “Token threshold that triggers automatic history compaction (unset uses model defaults).”, }, { key: “oss_provider”, type: “lmstudio | ollama”, description: “Default local provider used when running with `--oss` (defaults to prompting if unset).”, }, { key: “approval_policy”, type: “untrusted | on-failure | on-request | never”, description: “Controls when Codex pauses for approval before executing commands.”, }, { key: “sandbox_mode”, type: “read-only | workspace-write | danger-full-access”, description: “Sandbox policy for filesystem and network access during command execution.”, }, { key: “sandbox_workspace_write.writable_roots”, type: “array”, description: ‘Additional writable roots when `sandbox_mode = "workspace-write"`.’, }, { key: “sandbox_workspace_write.network_access”, type: “boolean”, description: “Allow outbound network access inside the workspace-write sandbox.”, }, { key: “sandbox_workspace_write.exclude_tmpdir_env_var”, type: “boolean”, description: “Exclude `$TMPDIR` from writable roots in workspace-write mode.”, }, { key: “sandbox_workspace_write.exclude_slash_tmp”, type: “boolean”, description: “Exclude `/tmp` from writable roots in workspace-write mode.”, }, { key: “notify”, type: “array”, description: “Command invoked for notifications; receives a JSON payload from Codex.”, }, { key: “check_for_update_on_startup”, type: “boolean”, description: “Check for Codex updates on startup (set to false only when updates are centrally managed).”, }, { key: “feedback.enabled”, type: “boolean”, description: “Enable feedback submission via `/feedback` across Codex surfaces (default: true).”, }, { key: “instructions”, type: “string”, description: “Reserved for future use; prefer `experimental_instructions_file` or `AGENTS.md`.”, }, { key: “developer_instructions”, type: “string”, description: “Additional developer instructions injected into the

session (optional).”, }, { key: “compact_prompt”, type: “string”, description: “Inline override for the history compaction prompt.”, }, { key: “experimental_instructions_file”, type: “string (path)”, description: “Experimental replacement for built-in instructions instead of `AGENTS.md`.”, }, { key: “experimental_compact_prompt_file”, type: “string (path)”, description: “Load the compaction prompt override from a file (experimental).”, }, { key: “mcp_servers..command”, type: “string”, description: “Launcher command for an MCP stdio server.”, }, { key: “mcp_servers..args”, type: “array”, description: “Arguments passed to the MCP stdio server command.”, }, { key: “mcp_servers..env”, type: “map<string,string>”, description: “Environment variables forwarded to the MCP stdio server.”, }, { key: “mcp_servers..env_vars”, type: “array”, description: “Additional environment variables to whitelist for an MCP stdio server.”, }, { key: “mcp_servers..cwd”, type: “string”, description: “Working directory for the MCP stdio server process.”, }, { key: “mcp_servers..url”, type: “string”, description: “Endpoint for an MCP streamable HTTP server.”, }, { key: “mcp_servers..bearer_token_env_var”, type: “string”, description: “Environment variable sourcing the bearer token for an MCP HTTP server.”, }, { key: “mcp_servers..http_headers”, type: “map<string,string>”, description: “Static HTTP headers included with each MCP HTTP request.”, }, { key: “mcp_servers..env_http_headers”, type: “map<string,string>”, description: “HTTP headers populated from environment variables for an MCP HTTP server.”, }, { key: “mcp_servers..enabled”, type: “boolean”, description: “Disable an MCP server without removing its configuration.”, }, { key: “mcp_servers..startup_timeout_sec”, type: “number”, description: “Override the default 10s startup timeout for an MCP server.”, }, { key: “mcp_servers..startup_timeout_ms”, type: “number”, description: “Alias for `startup_timeout_sec` in milliseconds.”, }, { key: “mcp_servers..tool_timeout_sec”, type: “number”, description: “Override the default 60s per-tool timeout for an MCP server.”, }, { key: “mcp_servers..enabled_tools”, type: “array”, description: “Allow list of tool names exposed by the MCP server.”, }, { key: “mcp_servers..disabled_tools”, type: “array”, description: “Deny list applied after `enabled_tools` for the MCP server.”, }, { key: “features.unified_exec”, type: “boolean”, description: “Use the unified PTY-backed exec tool (beta).”, }, { key: “features.shell_snapshot”, type: “boolean”, description: “Snapshot shell environment to speed up repeated commands (beta).”, }, { key: “features.apply_patch_freeform”, type: “boolean”, description: “Expose the freeform `apply_patch` tool (experimental).”, }, { key: “features.web_search_request”, type: “boolean”, description: “Allow the model to issue web searches (stable).”, }, { key: “features.shell_tool”, type: “boolean”, description: “Enable the default `shell` tool for running commands (stable; on by default).”, }, { key: “features.exec_policy”, type: “boolean”, description: “Enforce rules checks for `shell`/`unified_exec` (experimental; on by default).”, }, { key: “features.experimental_windows_sandbox”, type: “boolean”, description: “Run the Windows restricted-token sandbox (experimental).”, }, { key: “features.elevated_windows_sandbox”, type: “boolean”, description: “Enable the elevated Windows sandbox pipeline (experimental).”, }, { key: “features.remote_compaction”, type: “boolean”, description: “Enable remote compaction (ChatGPT auth only; experimental; on by default).”, }, { key: “features.remote_models”, type: “boolean”, description: “Refresh remote model list before showing readiness (experimental).”, }, { key: “features.powershell_utf8”, type: “boolean”, description: “Force PowerShell UTF-8 out-

put (defaults to true).”, }, { key: “features.tui2”, type: “boolean”, description: “Enable the TUI2 interface (experimental).”, }, { key: “model_providers..name”, type: “string”, description: “Display name for a custom model provider.”, }, { key: “model_providers..base_url”, type: “string”, description: “API base URL for the model provider.”, }, { key: “model_providers..env_key”, type: “string”, description: “Environment variable supplying the provider API key.”, }, { key: “model_providers..env_key_instructions”, type: “string”, description: “Optional setup guidance for the provider API key.”, }, { key: “model_providers..experimental_bearer_token”, type: “string”, description: “Direct bearer token for the provider (discouraged; use `env_key`).”, }, { key: “model_providers..requires_openai_auth”, type: “boolean”, description: “The provider uses OpenAI authentication (defaults to false).”, }, { key: “model_providers..wire_api”, type: “chat | responses”, description: “Protocol used by the provider (defaults to `chat` if omitted).”, }, { key: “model_providers..query_params”, type: “map<string,string>”, description: “Extra query parameters appended to provider requests.”, }, { key: “model_providers..http_headers”, type: “map<string,string>”, description: “Static HTTP headers added to provider requests.”, }, { key: “model_providers..env_http_headers”, type: “map<string,string>”, description: “HTTP headers populated from environment variables when present.”, }, { key: “model_providers..request_max_retries”, type: “number”, description: “Retry count for HTTP requests to the provider (default: 4).”, }, { key: “model_providers..stream_max_retries”, type: “number”, description: “Retry count for SSE streaming interruptions (default: 5).”, }, { key: “model_providers..stream_idle_timeout_ms”, type: “number”, description: “Idle timeout for SSE streams in milliseconds (default: 300000).”, }, { key: “model_reasoning_effort”, type: “minimal | low | medium | high | xhigh”, description: “Adjust reasoning effort for supported models (Responses API only; `xhigh` is model-dependent).”, }, { key: “model_reasoning_summary”, type: “auto | concise | detailed | none”, description: “Select reasoning summary detail or disable summaries entirely.”, }, { key: “model_verbosity”, type: “low | medium | high”, description: “Control GPT-5 Responses API verbosity (defaults to `medium`).”, }, { key: “model_supports_reasoning_summaries”, type: “boolean”, description: “Force Codex to send reasoning metadata even for unknown models.”, }, { key: “shell_environment_policy.inherit”, type: “all | core | none”, description: “Baseline environment inheritance when spawning subprocesses.”, }, { key: “shell_environment_policy.ignore_default_excludes”, type: “boolean”, description: “Keep variables containing KEY/SECRET/TOKEN before other filters run.”, }, { key: “shell_environment_policy.exclude”, type: “array”, description: “Glob patterns for removing environment variables after the defaults.”, }, { key: “shell_environment_policy.include_only”, type: “array”, description: “Whitelist of patterns; when set only matching variables are kept.”, }, { key: “shell_environment_policy.set”, type: “map<string,string>”, description: “Explicit environment overrides injected into every subprocess.”, }, { key: “shell_environment_policy.experimental_use_profile”, type: “boolean”, description: “Use the user shell profile when spawning subprocesses.”, }, { key: “project_root_markers”, type: “array”, description: “List of project root marker filenames; used when searching parent directories for the project root.”, }, { key: “project_doc_max_bytes”, type: “number”, description: “Maximum bytes read from `AGENTS.md` when building project instructions.”, }, { key: “project_doc_fallback_file-

names", type: "array", description: "Additional filenames to try when `AGENTS.md` is missing.", }, { key: "profile", type: "string", description: "Default profile applied at startup (equivalent to `--profile`).", }, { key: "profiles..*", type:"various", description: "Profile-scoped overrides for any of the supported configuration keys.", }, { key: "profiles..include_apply_patch_tool", type: "boolean", description: "Legacy name for enabling freeform apply_patch; prefer `[features].apply_patch_freeform`.", }, { key: "profiles..experimental_use_unified_exec_tool", type: "boolean", description: "Legacy name for enabling unified exec; prefer `[features].unified_exec`.", }, { key: "profiles..experimental_use_freeform_apply_patch", type: "boolean", description: "Legacy name for enabling freeform apply_patch; prefer `[features].apply_patch_freeform`.", }, { key: "profiles..oss_provider", type: "lmstudio | ollama", description: "Profile-scoped OSS provider for `--oss` sessions.", }, { key: "history.persistence", type: "save-all | none", description: "Control whether Codex saves session transcripts to history.jsonl.", }, { key: "tool_output_token_limit", type: "number", description: "Token budget for storing individual tool/function outputs in history.", }, { key: "history.max_bytes", type: "number", description: "If set, caps the history file size in bytes by dropping oldest entries.", }, { key: "file_opener", type: "vscode | vscode-insiders | windsurf | cursor | none", description: "URI scheme used to open citations from Codex output (default: `vscode`).", }, { key: "otel.environment", type: "string", description: "Environment tag applied to emitted OpenTelemetry events (default: `dev`).", }, { key: "otel.exporter", type: "none | otlp-http | otlp-grpc", description: "Select the OpenTelemetry exporter and provide any endpoint metadata.", }, { key: "otel.trace_exporter", type: "none | otlp-http | otlp-grpc", description: "Select the OpenTelemetry trace exporter and provide any endpoint metadata.", }, { key: "otel.log_user_prompt", type: "boolean", description: "Opt in to exporting raw user prompts with OpenTelemetry logs.", }, { key: "otel.exporter..endpoint", type: "string", description: "Exporter endpoint for OTEL logs.", }, { key: "otel.exporter..protocol", type: "binary | json", description: "Protocol used by the OTLP/HTTP exporter.", }, { key: "otel.exporter..headers", type: "map<string,string>", description: "Static headers included with OTEL exporter requests.", }, { key: "otel.trace_exporter..endpoint", type: "string", description: "Trace exporter endpoint for OTEL logs.", }, { key: "otel.trace_exporter..protocol", type: "binary | json", description: "Protocol used by the OTLP/HTTP trace exporter.", }, { key: "otel.trace_exporter..headers", type: "map<string,string>", description: "Static headers included with OTEL trace exporter requests.", }, { key: "otel.exporter..tls.ca-certificate", type: "string", description: "CA certificate path for OTEL exporter TLS.", }, { key: "otel.exporter..tls.client-certificate", type: "string", description: "Client certificate path for OTEL exporter TLS.", }, { key: "otel.exporter..tls.client-private-key", type: "string", description: "Client private key path for OTEL exporter TLS.", }, { key: "otel.trace_exporter..tls.ca-certificate", type: "string", description: "CA certificate path for OTEL trace exporter TLS.", }, { key: "otel.trace_exporter..tls.client-certificate", type: "string", description: "Client certificate path for OTEL trace exporter TLS.", }, { key: "otel.trace_exporter..tls.client-private-key", type: "string", description: "Client private key path for OTEL trace exporter TLS.", }, { key: "tui", type: "table", description: "TUI-specific options such as enabling inline desktop notifications.", }, { key: "tui.notifications", type: "boolean | array", de-

scription: “Enable TUI notifications; optionally restrict to specific event types.”, }, { key: “tui.animations”, type: “boolean”, description: “Enable terminal animations (welcome screen, shimmer, spinner) (default: true).”, }, { key: “tui.show_tooltips”, type: “boolean”, description: “Show onboarding tooltips in the TUI welcome screen (default: true).”, }, { key: “tui.scroll_events_per_tick”, type: “number”, description: “Wheel event density used to normalize TUI2 scrolling.”, }, { key: “tui.scroll_wheel_lines”, type: “number”, description: “Lines per wheel notch for TUI2 scrolling.”, }, { key: “tui.scroll_trackpad_lines”, type: “number”, description: “Baseline trackpad scroll sensitivity for TUI2.”, }, { key: “tui.scroll_trackpad_accel_events”, type: “number”, description: “Trackpad events required to gain +1x acceleration.”, }, { key: “tui.scroll_trackpad_accel_max”, type: “number”, description: “Maximum acceleration multiplier for trackpad scrolling.”, }, { key: “tui.scroll_mode”, type: “auto | wheel | trackpad”, description: “Scroll interpretation mode for TUI2.”, }, { key: “tui.scroll_wheel_tick_detect_max_ms”, type: “number”, description: “Auto-mode wheel tick detection threshold (ms).”, }, { key: “tui.scroll_wheel_like_max_duration_ms”, type: “number”, description: “Auto-mode wheel fallback duration threshold (ms).”, }, { key: “tui.scroll_invert”, type: “boolean”, description: “Invert mouse scroll direction in TUI2.”, }, { key: “hide_agent_reasoning”, type: “boolean”, description: “Suppress reasoning events in both the TUI and `codex exec` output.”, }, { key: “show_raw_agent_reasoning”, type: “boolean”, description: “Surface raw reasoning content when the active model emits it.”, }, { key: “disable_paste_burst”, type: “boolean”, description: “Disable burst-paste detection in the TUI.”, }, { key: “windows_wsl_setup_acknowledged”, type: “boolean”, description: “Track Windows onboarding acknowledgement (Windows only).”, }, { key: “chatgpt_base_url”, type: “string”, description: “Override the base URL used during the ChatGPT login flow.”, }, { key: “cli_auth_credentials_store”, type: “file | keyring | auto”, description: “Control where the CLI stores cached credentials (file-based auth.json vs OS keychain).”, }, { key: “mcp_oauth_credentials_store”, type: “auto | file | keyring”, description: “Preferred store for MCP OAuth credentials.”, }, { key: “experimental_use_unified_exec_tool”, type: “boolean”, description: “Legacy name for enabling unified exec; prefer `[features].unified_exec` or `codex --enable unified_exec`.”, }, { key: “experimental_use_freeform_apply_patch”, type: “boolean”, description: “Legacy name for enabling freeform apply_patch; prefer `[features].apply_patch_freeform` or `codex --enable apply_patch_freeform`.”, }, { key: “include_apply_patch_tool”, type: “boolean”, description: “Legacy name for enabling freeform apply_patch; prefer `[features].apply_patch_freeform`.”, }, { key: “projects..trust_level”, type: “string”, description: ‘Mark a project or worktree as trusted or untrusted (`"trusted"` | `"untrusted"`).’, }, { key: “notice.hide_full_access_warning”, type: “boolean”, description: “Track acknowledgement of the full access warning prompt.”, }, { key: “notice.hide_world_writable_warning”, type: “boolean”, description: “Track acknowledgement of the Windows world-writable directories warning.”, }, { key: “notice.hide_rate_limit_model_nudge”, type: “boolean”, description: “Track opt-out of the rate limit model switch reminder.”, }, { key: “notice.hide_gpt5_1_migration_prompt”, type: “boolean”, description: “Track acknowledgement of the GPT-5.1 migration prompt.”, }, { key: “notice.hide_gpt-5.1-codex-max_migration_prompt”, type:

“boolean”, description: “Track acknowledgement of the gpt-5.1-codex-max migration prompt.”, }, { key: “notice.model_migrations”, type: “map<string,string>”, description: “Track acknowledged model migrations as old->new mappings.”, }, { key: “forced_login_method”, type: “chatgpt | api”, description: “Restrict Codex to a specific authentication method.”, }, { key: “forced_chatgpt_workspace_id”, type: “string (uuid)”, description: “Limit ChatGPT logins to a specific workspace identifier.”, },]} client:load />

`requirements.toml`

`requirements.toml` is an admin-enforced configuration file that constrains security-sensitive settings users can’t override. For details, locations, and examples, see Admin-enforced requirements.

<ConfigTable options={[{ key: “allowed_approval_policies”, type: “array”, description: “Allowed values for `approval_policy`.”, }, { key: “allowed_sandbox_modes”, type: “array”, description: “Allowed values for `sandbox_mode`.”, },]} client:load />