TLAMATINI

Complete Architecture Atlas

eXtended Artificial Intelligence, Humanly Tempered · "one who knows"

Created by **Angela López Mendoza** · XAIHT

Release v1.92.4 · commit 52e97879 (2026-11-08) · atlas generated 2026-11-08 from application files tracked in git; database history is summarized

1,638 listed application files	**599,184** lines of text	**716** Python files	**87** workflow agent types
68 Multi-Turn chat-agent tools	**192 + 5** HTTP routes + WebSocket routes	**21** Django models	**221** migrations
64 / 29 first-party JS / CSS files	**29** SKILL.md packages	**6,386** test functions in 208 files	**244** Markdown documents

Lines of text by architectural layer



- Presentation (browser) — 119 files, 47,545 lines
- Process & transport — 36 files, 22,230 lines
- Intelligence core — 87 files, 38,023 lines
- Integration surfaces — 81 files, 22,128 lines
- Agent pool — 258 files, 119,670 lines
- Persistence & state — 5 files, 894 lines
- Platform & safety — 21 files, 6,489 lines
- Build, release & install — 58 files, 24,533 lines
- Verification — 398 files, 146,898 lines
- Knowledge & governance — 225 files, 55,683 lines
- Vendored third-party — 350 files, 115,091 lines

File-type markers on every tile

- Python
- JavaScript
- CSS
- HTML
- Markdown
- YAML/JSON
- PS1/BAT/CMD
- Prompts/flows
- C++/Arduino
- Text/other
- Binary/media

How to read this atlas

- **Part A (sheets 2–15)** draws the architecture: runtime topology, request lifecycle, intelligence core, agents, integrations, frontend, data, build/release, verification and a module dependency matrix. File names in blue are links to the index.
- **Part B (sheets 16–24)** is the file atlas: every listed application file is a tile inside the cluster where it lives architecturally. Marker = file type, right number = lines (or size for binaries).
- **Part C** is the file index: every file with its ID, path, size and a description extracted from the file itself (docstring, header comment, heading or parsed structure).
- Click a tile to open its index entry; click an ID in the index to jump back to the tile. PDF bookmarks list every sheet and cluster.

SHEET INDEX

Part A — architecture overview

02 Runtime topology — processes, ports and stores
03 Request lifecycle — from the composer to the rendered answer
04 Live canvas editor and Interactor
05 Live Editor and Terminater configuration
06 Torrenter and release delivery
07 Matching releases and isolated builds
08 Intelligence core — five layers, the Multi-Turn executor and the Model Brain
09 Agent ecosystem — 87 workflow agent types and the flow pipeline
10 Integration surfaces — ACPX, skills, external MCPs and bridges
11 Frontend module map — what each page loads, in order
12 Data model and state — 21 Django models, 221 migrations, WAL-safe copies
13 Build, release, install and self-update pipeline
14 Verification and knowledge — how Tlamatini is tested and documented
15 Python module dependency matrix (DSM)

Part B — file atlas

16 Atlas I — Server core: process, web tier and intelligence
17 Atlas II — Integration surfaces, platform and state
18 Atlas IV — Presentation layer: first-party frontend
19 Atlas V — Vendored third-party frontend and PDF.js
20 Atlas VI — Agent pool: every workflow agent template
21 Atlas VI — Agent pool: every workflow agent template (continued 2)
22 Atlas VII — Verification: unit, contract, visible and release tests
23 Atlas VIII — Build, release, install, security and repository root
24 Atlas IX — Knowledge and governance

Part C — file index (1638 entries)

Runtime topology — processes, ports and stores

One Python process serves the browser pages over HTTP and 5 WebSocket routes, starts two built-in MCP servers, and drives pool agents, ACPX children and external MCP servers.

SHEET 02

Chat page

- agent_page.html: composer, toolbar modes (Multi-Turn, Exec report, ACPX, Ask Execs, Step-by-Step, Compact mode, Self-modify), canvas, avatar
- Boot and state: agent_page_init.js, agent_page_state.js, agent_page_ui.js, agent_page_layout.js
- Messages: agent_page_chat.js (Copy/Drop, Create Flow, Exec Report)
- Canvas: agent_page_canvas.js, agent_page_markdown.js, agent_page_pdf.js
- Voice and avatar: chat_dictation.js, mic_settings.js, avatar.js
- Gauges: context_gauge.js, model_capacity.js, self_modify_switch.js

HTTP, WS

Agentic Control Panel

- agentic_control_panel.html + agentic_control_panel.js
- Canvas engine: acp-canvas-core.js, acp-canvas-undo.js, acp-undo-manager.js, acp-editor-tools.js
- Flow I/O: acp-file-io.js (.flw), acp-flow-snapshot.js → /agent/compile_flow/
- Run control: acp-control-buttons.js, acp-running-state.js, acp-session.js, acp-validate.js

HTTP

Prompt Flow Panel

- prompt_flow_panel.html: diagram editor for chains of prompts saved as .fpmt
- prompt-flow-panel-model.js (document contract) and prompt-flow-panel.js (editor, playback)
- Shared canvas mechanics: flow-canvas-interactions.js, flow_canvas.css

WS

Entry pages, admin and PDF frame

- login.html, welcome.html + welcome_enter_default.js
- Django admin skin: base_site.html + admin_theme.css
- PDF iframe: pdf/canvas.html + pdf_canvas_viewer.js + vendored PDF.js

HTTP

Shared shell on every page

- Dismissal policy and themed popups: dialog_policy.js, dialog_theme.css
- Theme and splitters: ui_tokens.css, ui_shell.css, ui_splitters.css, ui_splitters.js
- Smart file drop: file_drop_router.js; bulk toggle: checkbox_bulk_toggle.js

TLAMATINI SERVER — manage.py → Django + Channels on Daphne (ASGI), port django_port (default 8000)

Process bootstrap

- manage.py: tees stdout/stderr into tlamatini.log, console shield, coloured console, django_port, Temp pin, post-update DB swap and migrate
- tlamatini/settings.py (SQLite WAL, WhiteNoise), tlamatini/asgi.py, tlamatini/middleware.py (per-user log tags), tlamatini/logging_filters.py
- agent/apps.py AgentConfig.ready(): rebuilds the Agent table, starts the MCP servers, boot_skills(), Tier-3 reaper, companion manifest

HTTP surface

- tlamatini/urls.py → agent/urls.py: 192 routes
- agent/views.py (pages, agents, flows, DB, prompts, skills, external MCPs), flow_file_views.py, model_brain_views.py, pdf_context_views.py, usage_views.py, agent/admin.py

WebSocket surface (routing.py)

- Routes declared in agent/routing.py
- ws/agent/ → agent/consumers.py AgentConsumer: chat, toggles, Drop, Compact and Self-modify switches, Ask-Execs replies
- ws/prompt-flow-panel/ → prompt_flow_panel_consumer.py (one run per connection)
- ws/chat-voice/ → chat_voice_consumer.py (direct microphone)

RAG and answer chains

- rag/interface.py ask_rag() → rag/factory.py builds the chain; sidecars inject system and file context
- rag/chains/basic.py, rag/chains/history_aware.py, rag/chains/unified.py
- Retrieval: rag/retrieval.py, rag/loaders.py, rag/splitters.py, rag/binary_guard.py (FAISS + BM25)

Multi-Turn agent

- mcp_agent.py: capability-aware tool-loop executor
- global_execution_planner.py, capability_registry.py, self_healing.py, exec_permission.py
- context_fitter.py, compact_mode.py, model_brain.py, model_brain_chat.py

Tool surface

- agent/tools.py: direct tools and one chat_agent_* launcher per wrapped agent (chat_agent_registry.py: 68 specs; chat_agent_runtime.py)
- acpx/tools.py: 12 ACPX/Skill tools; external_mcp_manager.py: 10 supervisors + ext__<server>__<tool>
- agent_verdict.py: deterministic SUCCESS/FAILED from each agent's INI_SECTION

Flow services

- services/flow_spec.py → services/flow_compiler.py with services/agent_contracts.py and services/agent_paths.py
- services/response_parser.py: answer cleanup and Exec Report tables
- services/prompt_flow_panel.py: .fpmt validator and interpreter

Platform guards

- path_guard.py (Temp/Templates policy), orphan_reaper.py (three-tier reaper), command_watchdog.py
- log_identity.py ([a3] user tags), cancellation.py, global_state.py

read / write

WS

System-Metrics MCP · WebSocket :8765

- mcp_system_server.py ↔ mcp_system_client.py; sidecar chain_system_lcel.py

gRPC

Files-Search MCP · gRPC :50051

- mcp_files_search_server.py, filesearch.proto → filesearch_pb2.py, filesearch_pb2_grpc.py; sidecar chain_files_search_lcel.py, bounded by context_sidecar_timeout.py

HTTP

LLM back-ends

- Ollama local and cloud (chat, embeddings, vision); sampling per model from model_profiles.json
- Anthropic Claude: claude_opus_client.py; images: imaging/image_interpreter.py

spawn

Pool agent processes

- 87 templates in agent/agents/<type>/, copied to agents/pools/<session>/<type>_N (canvas) or _chat_runs_ (chat)
- Each run: <type>.py + config.yaml, PID file, reanim* state, INI_SECTION_<TYPE> output

stdio

ACPX child CLIs

- claude, codex, gemini, cursor, qwen, kiro, kimi, copilot… spawned by acpx/runtime.py via acpx/windows_spawn.py and win_shim.py; verdict acpx/child_health.py

connect

External MCP servers

- stdio, streamable-http, sse, websocket; at most 5 active; catalog external_mcps.json; defaults external_mcp_defaults.py; runtimes runtime_provisioner.py

socket

Voice worker

- whisperer/chat_worker.py over an authenticated loopback socket (chat_voice_runtime.py, chat_voice_settings.py)

Root stdio MCP server

- tlamatini_mcp_server.py + tlamatini_acpx.py expose every pool agent and ACPX to external MCP clients (Claude Code, Kimi): .mcp.json

STORAGE AND STATE

SQLite (WAL)

- db.sqlite3: 21 models in agent/models.py, 221 migrations; WAL-safe backup and Set-DB by sqlite_copy.py

User state and configuration

- config.json (agent/config.json template), external_mcps.json, contacts, preserved_user_state.json

Working directories

- <app>/Temp (all scratch) and <app>/Templates (scaffolded projects) via path_guard.py; agents/pools/ session pools

Logs

- tlamatini.log (truncated at start, per-user tags), each pool agent's own log with INI_SECTION blocks

Per-user machine stores

- %LOCALAPPDATA%\Tlamatini: private runtimes, memory knowledge graph, learned model profiles, setup logs

Request lifecycle — from the composer to the rendered answer

Twelve steps of one chat message in Multi-Turn mode, by the layer that performs them. Every application file named is clickable.

SHEET 03

Browser

1 · Compose and send
`agent_page_init.js` sends the prompt and the mode flags over `ws/agent/`

7 · Ask Execs
`exec_permission.py` broker → Proceed/Deny dialog (`agent_page_dialogs.js`); Deny halts the chain

12 · Render
`agent_page_chat.js` draws the card; `context_gauge.js` ring; `avatar.js` speaks; Create Flow → `/agent/flow_from_tool_calls/`

Channels consumer

2 · Receive
`agent/consumers.py` binds the user tag (`log_identity.py`), saves the prompt (`AgentMessage`), queues retrieval on a worker thread

11 · Broadcast
`agent_message` with `message_id` and `tool_calls_log`; Tier-2 reaper; `context_governor.py`; `usage_tracking.py`

RAG interface and chains

3 · Decide context
Internet check (`inet_determiner.py`); `rag/interface.py ask_rag()`; last 8 messages from `chat_history_loader.py`

4 · Build chain
`rag/factory.py` picks basic, history-aware or unified; patched `invoke()` pulls sidecar context (`chain_system_lcel.py`, `chain_files_search_lcel.py`)

Multi-Turn executor

5 · Plan and fit
`global_execution_planner.py` + `capability_registry.py`; ACPX filter; `fit_request` (`context_fitter.py`, `compact_mode.py`); `model_brain.py`

6 · Tool loop
`mcp_agent.py` executor; each model step through `self_healing.py` (watchdog, tactic ladder); dedup and quota guards

Tools and agents

8 · Run tools
`agent/tools.py` → `chat_agent_runtime.py` copies and starts a pool agent; `acp_*` children; `ext__*` remote tools

9 · Judge
Agent prints `INI_SECTION_<TYPE>`; `agent_verdict.py` rules SUCCESS/FAILED; Tier-1 `orphan_reaper.py`

Response and persistence

10 · Compose answer
`services/response_parser.py` strips `END-RESPONSE`, appends the Exec Report and any denial banner, then saves the row

Mode gates

Multi-Turn off: the legacy one-shot answer. ACPX off: `filter_acpx_tools` strips the 12 ACPX/Skill tools. Exec report and Ask Execs act only inside Multi-Turn. Step-by-Step paces one action per reply; Compact mode binds only the ticked tools.

Never hang, never discard, never lie

Each model attempt has a watchdog; distinct recovery tactics follow; a graceful answer is built from the work already done, and a recovery preamble tells the user what happened. Only Cancel (`cancellation.py`) or an exhausted ladder stops it.

What returns to the browser

Answer HTML; Exec Report tables (one per agent kind); `tool_calls_log` for Create Flow; context-gauge frames with Ollama's real token count; Compact and Self-modify state frames; message ids for Drop.

Live canvas editor and Interactor

Editable source, native persistence and tutoring share one document authority. Interactor is a module/tool, not a workflow agent.

SHEET 04

Editable document

- `agent_page_editing.js` owns the CodeMirror buffer, dirty baseline, encoding and save snapshots.
- `agent_page_canvas.js` owns Open/Refresh and cancellation; Markdown Preview/Source share the current buffer.
- Text/code/JSON are editable. HTML/SVG stay inert source. PDFs remain read-only and export original bytes.
- Save, Save as, Undo/Redo, Find/replace, Wrap; Ctrl/Cmd+S and Ctrl/Cmd+Shift+S.

Learner control and latency

- Live editor opt-in; first read exact unsaved text and selection, then type a short segment.
- guide hands the next field to the learner and ends mutations for that turn. Continue starts fresh chat/read.
- Prepared choices match locally: Tab accepts, Ctrl+Right takes one word, Escape dismisses. No per-key model call.
- User input/selection, Stop, cancellation or disconnect interrupts and preserves the typed prefix.

Visibility and layout

- A hidden/occluded tab pauses the same command for up to 120 extra seconds; resume at the next character.
- No automatic replay after a partial edit. Preview or hiding the canvas returns control.
- Automatic layout reserves canvas width; editor fills available height. Manual divider choices remain valid.

Authenticated live bridge

- `interactor/consumer.py` serves ws/interactor/ with exact Origin/Host and session authentication.
- `interactor/hub.py` binds server-issued tab IDs to account/session; ContextVar carries identity into tool workers.
- Bounded command queues, IDs, deadlines and document/revision checks reject stale work. Full-buffer receipts verify edits.
- Live tutoring: 262,144 UTF-16 units; a smaller tool-result budget can refuse content, never silently truncate it.

Native file persistence

- `canvas_file_views.py` and `native_dialogs.py` provide local Save As outside Downloads.
- Authenticated CSRF-protected loopback requests use account/session-bound capabilities; browser filenames are not host-path authority.
- Compare full disk revision, stage and flush on the same volume, recheck, then replace. Preserve BOM/encoding, line endings and Windows metadata.
- 64 MiB text/native-save limit. Save cancellation/failure keeps dirty edits. DB backup/restore is unchanged.

Tool and release carriage

- Migration 0218 registers the Tool and catalog example, not an Agent. tool_gate_table drives binding, Compact and costs.
- Ask Execs gates new/save. New unnamed-path work uses a unique Temp/interactor directory; manual files need user-granted write access.
- Password input is not command ownership; an email field or DONE reply does not authorize outbound notification.
- Live Editor accepts 40-250 WPM and Terminater accepts 20-250 WPM. Linked speeds use 40-250.

Live Editor and Terminater configuration

One account schema controls real actions; local appearance and a shared speech queue complete the user experience.

SHEET 05

Dialog and draft

- Config / canvas Settings / terminal Settings open `interaction_settings.html` on chat, ACP and PFP.
- `interaction_settings.js` builds Live Editor, Terminater and Voice fields from the server schema.
- Presets, reset and preview edit a draft; Save applies, Cancel/Escape discard. Stop, mute and tab access are immediate.
- `interaction_settings.css` keeps tabs, scrolling and footer controls inside the viewport.

Account settings and revisions

- `interaction_settings.py` validates ranges/types and serves authenticated CSRF-protected /agent/interaction-settings/.
- Migration 0220 adds InteractionSettings: one user, JSON preferences, optimistic revision.
- Each action reads a committed snapshot. Conflicting saves ask the user to reload; appearance stays account/browser-local.

Real editor behavior

- `interactor/tool.py` returns teaching-detail guidance and sends saved pacing, pause and narration with the action.
- `interactor/bridge.js` keeps document/revision checks and user-first cancellation; guide applies suggestions/ghost choices.
- `agent_page_editing.js` and the offline CodeMirror bundle apply font, wrap and line numbers; manual scrolling suspends following.

One terminal dispatch policy

- `terminal_hub.py` attaches account preference, presentation and result callbacks to the stdlib TerminalHub engine.
- Chat, ACP, PFP and browser actions share it. New flows can use_my_defaults; old explicit flow values remain.
- User-only Enter rejects automatic submission and control characters, including aliases/base64. User keys still work.
- Shell/advisor defaults affect new terminals. Command wait, no-output and handoff windows stay distinct; timeout does not kill.

Narration receipts and cancellation

- `interaction_narration.py` selects one focused visible account tab; receipts are bound to its exact connection.
- Before-typing wait is bounded to 120 seconds. Missing listener, Stop or disconnect cancels; reconnect never replays.
- `voice_runtime.js` shares one queue with avatar speech; `interaction_speech.js` offers explicit skip/stop when voice is unavailable.
- Mute and microphone listening win. Secret-marked text and handoff values are not narrated; results speak status only.

Delivery and dated evidence

- `build_runtime_assets.py`, `copy_source_assets.py`, frozen imports, templates and rebuilt editor manifest travel together.
- October 23: 17 visible browser checks passed; real OS speech completed before the first terminal key.
- Implementation regressions: 929/931 passed; two baseline README assertions were corrected by the documentation sweep.
- Existing tag a771d27c. No new executable build or installed acceptance is claimed.

Torrenter and release delivery

Latest released version v1.92.4

SHEET 06

1. Sign in and choose the account

- Chat, the Agentic Control Panel and XT open the same account queue. Migration 0221 registers Torrenter.
- Django authenticates the browser. The controller obtains a scoped credential from Tlachihuani Jobs.

2. Inspect the source and select files

- Inspect the magnet or torrent. Keep its inspection ID until the metadata and file IDs are ready.
- A fifth magnet can queue behind four active metadata lookups. Poll the same ID.

3. Submit and follow the job

- Submit selected IDs and the metadata revision. Reuse the operation key on an uncertain retry.
- An ACP node can continue on admission or wait for verified publication. Pausing preserves the wait.

4. Verify the delivered copy

- Read the path and SHA-256 from the published manifest. The destination is Windows Downloads or the edition's content_generated folder.
- Private references and activity belong in TorrenterData. Erasing files requires the matching preview approval in XT.

Distribute an update

- The runtime receipt identifies Uninstaller.exe. The swapper installs the verified replacement and preserves an existing data.keys.
- Ship the workspace HTML, JavaScript and CSS with the controller. Include them in the self-modify snapshot.

Build and run on Windows

- Isolate build environments by requirements fingerprint. The OS lock covers preparation and use.
- Executer preserves shell assignments such as PLAYWRIGHT_BROWSERS_PATH and applies the configured command timeout.

Matching releases and isolated builds

Latest released version v1.92.4 — English and Spanish

SHEET 07

1. Predict before compiling

- The combo launcher previews each real resolver and version emitter in memory.
- Compare the application-reported number, not tag placement. Pin one –version for both wrappers.

2. Build and archive both editions

- Private or Public mode runs both complete wrappers concurrently with –self-modify.
- Verify new ZIPs and SHA-256; archive by edition/type/start minute. Preserve a successful sibling on failure.

3. Identify the package in its log

- The generated version module carries release_type. First log lines show version and Private/Public.
- Development and unspecified builds are explicit; never infer type from credentials or folder names.

4. Recover within isolated caches

- Use application-owned Temp/build-env or separate edition caches beneath the release root.
- If setup fails without a healthy cache, stop before ambient NumPy cleanup or pip –user.

5. Open readable Markdown faithfully

- Occasional control characters no longer make a readable README.md binary. Preserve the original bytes.
- Strict BOM-aware decoding still refuses NUL, control-dominated content and malformed encodings.

Dated build evidence

- November 8: four real English/Spanish Private/Public archives and executable checks passed at 1.92.3.
- Integrity, public-data audits and 26 isolation tests passed. This document refresh is not installed acceptance.

Intelligence core — five layers, the Multi-Turn executor and the Model Brain

How context reaches the model, how each request is fitted to the model's real window, and how every model step is protected.

SHEET 08

1 · Persisted toggles

- `Mcp`, `Tool`, `Agent`, `Skill` rows (`agent/models.py`) loaded by `agent/consumers.py` into status flags; `compact_mode.py` rewrites them as one switch

2 · Runtime MCP services

- System-Metrics (`mcp_system_server.py`) and Files-Search (`mcp_files_search_server.py`), started by `agent/apps.py` or `startserver.py`

3 · Context sidecars

- `chain_system_lcel.py` and `chain_files_search_lcel.py` decide whether `system_context` or `files_context` is needed

4 · Answer chains

- `rag/chains/basic.py`, `rag/chains/history_aware.py`, `rag/chains/unified.py`; `rag/factory.py` patches `invoke()`; `rag/chains/base.py` callbacks

5 · Unified-agent tools

- `agent/tools.py` `get_mcp_tools()` + `tool_gate_table()`: direct, wrapped (`chat_agent_*`), ACPX and external-MCP tools

Chain selection

- `basic`: prompt only, no documents (`rag/chains/basic.py`)
- `history_aware`: RAG with reranking over the loaded context (`rag/chains/history_aware.py`)
- `unified`: tool-enabled agent chains (`rag/chains/unified.py`); Multi-Turn binds the full enabled tool surface
- Sidecars fetch System-Metrics or Files-Search context only when the question needs it

Ask-Execs allowlist

- Prompts only for tools in `_ASK_EXECS_REQUIRED_TOOLS` (`mcp_agent.py`): command and script runners; tier A data-changing (Deleter, Mover, File-Creator, Editor, De-Compresser, PDFer, LaTeXer); tier D remote/network (SCPer, Apirer, Nmapper, Discoverer, Crawler, NetSpeed-Calculator)
- Not gated on purpose: tier B messaging and tier C visible desktop/hardware agents
- Deny halts the whole chain and shows a red banner; pinned by `test_ask_execs_allowlist.py`

Request payload

- message, history, `multi_turn_enabled`, `acpx_enabled`, `ask_execs_enabled`, `exec_report_enabled`, `step_by_step_enabled` (whitelisted when `unified.py` rebuilds the payload)

fit_request

- `context_fitter.py` measures the prompt and tool schemas against the model's real window: FULL, COMPACT or STRICT (locked); switch state in `compact_mode.py` (`CompactState`)

System prompt

- `agent/prompt.pmt` + self-knowledge `agent/Tlamatini.md` (only when Self-modify is on and fits) via `rag/config.py`; Temp and Templates rules

Model step

- `self_healing.py` `SelfHealingInvoker` → `model_brain_chat.py` `BrainChatOllama` (reasoning kept inside the tool loop); stop sequences fitted by `ollama_stop.py`; text tool-calls recovered by `local_toolcall_parser.py`

loop

Tool call

- dedup and quota → Ask-Execs allowlist (`exec_permission.py`) → tool → Tier-1 reaper (`orphan_reaper.py`) → verdict (`agent_verdict.py`)

Result

- `output`, `tool_calls_log`, `exec_report_entries`, `exec_report_denied`, recovery preamble → `services/response_parser.py`

Self-healing tactic ladder

- Each model attempt runs under a watchdog (`unified_agent_llm_step_timeout_seconds`, default 80 s)
- Distinct tactics: plain retry, back-off, message-tail trim, plain-LLM fallback, up to `unified_agent_llm_step_max_tactics`
- Live status lines reach the chat; the answer carries a recovery preamble; finished work is kept, so Create Flow and the Exec Report survive

Verdict engine

- `agent_verdict.py` parses `INI_SECTION_<TYPE>` into a typed record; ordered rules decide SUCCESS or FAILED; the agent's self-report outranks the exit code
- Closed vocabulary of five disjoint status sets, pinned by `test_status_vocabulary.py`

Model Brain

- Sources in order: overrides › `model_profiles.json` › learned profile › Ollama `/api/show` › model defaults
- `model_brain.py`, `model_brain_views.py`; Auto-tuning dialog `model_brain_tuning.js`

Context governor

- `context_governor.py` pairs Ollama's `prompt_eval_count` with each request; `context_baseline.py` probes the next request at rest; ring in `context_gauge.js`

Retrieval stack

- FAISS + BM25 (`rag/retrieval.py`), metadata (`rag_enhancements.py`), `rag/loaders.py`, `rag/splitters.py`, `rag/binary_guard.py`, VRAM pre-flight `embedding_memory_guard.py`, `gpu_perf.py`

Documents and vision

- PDF context: `pdf_context.py`, `pdf_context_jobs.py`, `pdf_image_analysis.py`; `visual_error_reporting.py`; `imaging/converter.py`, `imaging/screenshot.py`

Usage, voice and web

- `usage_provider.py` (Ollama balance), `usage_tracking.py` (`UsageDaily`); dictation `chat_voice_runtime.py`; `web_search_llm.py`, `llm_timing.py`

Prompt Flow runtime

- `prompt_flow_panel_runtime.py`: own chain, history and embeddings per run; interpreter `services/prompt_flow_panel.py`

Compact mode switch

- Ticking it unticks every MCP, tool, agent and skill row except System-Metrics, Files-Search and Current-Time; unticking re-ticks every row
- STRICT: a model that cannot hold everything activated locks it ON; saved rows apply at once (`after_toggles_saved`)
- Toolbar box `model_capacity.js`; per-row token prices `compact_costs.js`

Self-modify switch

- Rendered only where self-modify is possible: a source run, or a `-self-modify` build
- ON sends `agent/Tlamatini.md` as self-knowledge when it fits; a small model shows it locked OFF (`self_modify_switch.js`)

Ollama limits handled

- At most four stop sequences per request (`ollama_stop.py`); the real context window is learned from Ollama's `prompt_eval_count` when a request is cut

Agent ecosystem — 87 workflow agent types and the flow pipeline

From a drawing on the canvas (or a chat answer) to running processes; every chip opens that agent's files in the atlas.

SHEET 09

Design
- ACP canvas (`acp-canvas-core.js`); FlowCreator AI (`flowcreator.py`, `agentic_skill.md`, `flow_catalog.json`); chat Create Flow

→ **Snapshot**
- `acp-flow-snapshot.js` builds `schemaVersion 2`; `.flw` files via `acp-file-io.js`

→ **Normalize**
- `services/flow_spec.py`: FlowNode, FlowConnection, FlowSpec

→ **Compile**
- `services/flow_compiler.py` wires slots per `services/agent_contracts.py`, redacts secrets; names from `services/agent_paths.py`

→ **Session pool**
- `agents/pools/<session>/<type>_N/config.yaml` + `interconnection-scheme.csv`; shared helpers `model_settings.py`

→ **Run**
- Starter → agents → Parametrizer (`INI_SECTION_<TYPE>`) → Ender → Cleaner; LEDs and logs polled by `acp-running-state.js`

→ **Watch**
- FlowHypervisor (`monitoring-prompt.pmt`) answers OK or ATTENTION NEEDED

WORKFLOW AGENTS BY CATEGORY

chip = agent · MT = also a Multi-Turn chat tool · number = tracked files

Control Agents (6)
start, stop, schedule and clean up flows

Cleaner	3	Croner	2
Ender	2	Sleeper MT	2
Starter	2	Stopper	2

Multi-Channel Bridges (1)
bridges between outside channels and Tlamatini

TeleTlamatini	2

Anatomy of a pool agent
- Each template: `<type>.py` + `config.yaml`; flat helper siblings travel with pool copies (PDFer, PPTXer, LaTeXer, Googler, Video-Analyzer…)
- Lifecycle: PID file while running; `AGENT_REANIMATED=1` resumes after pause with `reanim*` state; Ender clears it
- Connections: `source_agents` (watch), `target_agents` (start), special slots for OR/AND, Forker/Asker, Counter, Ender
- Pool agents import nothing from `agent.*`: 0 of 133 pool Python files do (checked)

Routing Agents (4)
branch on patterns, choices and counters

Asker MT	2	Counter	2
Forker	2	Raiser	2

Utility Agents (3)
parametrize, back up, design and supervise flows

FlowBacker	2	FlowCreator MT	6
Parametrizer	2		

Logic Gates (3)
join parallel branches

AND	2	Barrier	3
OR	2		

Cryptography Agents (3)
post-quantum CRYSTALS-Kyber

Kyber-Cipher MT	2	Kyber-DeCipher MT	2
Kyber-KeyGen MT	2		

Action Agents (58)
do the work: shell, files, web, cloud, devices, media, documents

Analyzer	2	Apirer MT	2
Arduiner MT	7	AudioPlayer MT	2
Blenderer MT	2	Camcorder MT	2
Crawler MT	2	De-Compresser MT	2
Deleter MT	2	Discoverer MT	2
Dockerer MT	2	Editor MT	2
ESP32er MT	8	ESPHomer MT	8
Executer MT	2	File-Creator MT	2
File-Extractor MT	2	File-Interpreter MT	2
Gitter MT	2	Globber MT	2
Googler	3	Grepper MT	2
Image-Interpreter MT	2	Instant Messaging Doctor MT	2
J-Decompiler MT	2	Jenkinser MT	2
Kalier MT	2	Keyboarder MT	3
Kuberneter MT	2	LaTeXer MT	6
MCP Doctor MT	2	Mongoxer MT	2
Mouser MT	3	Mover MT	2
NetSpeed-Calculator MT	2	Nmapper MT	2
PDFer MT	15	Playwrighter MT	2
PPTXer MT	15	Prompter MT	2
PSer MT	2	Pythonxer MT	2
Recorder MT	2	Reviewer	2
SCPer MT	2	Shoter MT	2
SQLer MT	2	SSHer MT	2
STM32er MT	2	Summarizer MT	2
Talker MT	2	Terminater MT	5
Torrenter MT	6	Unrealer MT	2
Video-Analyzer MT	5	VideoPlayer MT	2
Whisperer MT	4	Windower MT	2

Notification Agents (8)
watch, notify and message people

Emailer MT	2	FlowHypervisor	4
Monitor Netstat MT	2	Monitor-Log MT	2
Notifier MT	2	RecMailer MT	2
Telegrammer MT	7	Whatsapper MT	8

External Coding-Agent Driver (1)
external coding-agent CLIs from the canvas

ACPXer	2

Integration surfaces — ACPX, skills, external MCPs and bridges

Four ways Tlamatini reaches beyond its own process: coding-agent CLIs, procedural skills, any MCP server, and the outside world.

ACPX — external coding-agent CLIs

Registry and transports

- `acpx/agent_registry.py`: agent ids with `oneshot-prompt`, `json-acp`, `tui-repl`, `one-shot` transports; per-agent overrides from `config.json` (`acpx/config.py`)

LLM surface

- `acpx/tools.py`: `acp_doctor`, `acp_spawn`, `acp_send`, `acp_send_and_wait`, `acp_relay`, `acp_kill`, transcripts, sessions, `list_skills`, `invoke_skill`

Runtime

- `acpx/runtime.py`: reader thread, transport-aware idle rule, NDJSON transcripts (`acpx/session_store.py`); `acpx/permissions.py` gate

Twins

- Canvas node `acpxer.py` and the Django-free root `tlamatini_acpx.py` share the same verdict; Windows spawning by `acpx/windows_spawn.py` + `win_shim.py`

Delivery verdict

- `acpx/child_health.py`: PERMISSION_BLOCKED, AUTH_FAILED, NO_CREDIT, NO_OUTPUT… an exit code is one bit, so the child's output is judged

Tests

- `acpx/tests.py`: every failure string copied from real transcripts

Skills — SKILL.md packages

Harness

- `skills/registry.py` discovers packages, `skills/frontmatter.py` parses, `skills/validation.py` is the one validator, `skills/harness.py` returns a planning handoff, `skills/io_contract.py`, `skills/redaction.py`

Boot and admin

- `acpx/service.py` `boot_skills()` mirrors packages into `Skill` rows; ACPX-Skills menu `skills_dialog.js`

29 packages in agent/skills_pkg/

`acp_router`, `adding_external_mcp`, `code_review`, `create_new_agent`, `create_new_mcp`, `flow_making`, `github`, `gmail`, `hello_world`, `jira`, `kali_pentest`, `notion`, `roblox_studio`, `security_audit`, `setup_new_acpx_key`, `skill_creator`, `slack`, `summarize`, `tlamatini_allowed_hosts_tighten`, `tlamatini_csrf_exempt_audit`, `tlamatini_exec_report_row_adder`, `tlamatini_flow_from_objective`, `tlamatini_flw_doctor`, `tlamatini_new_acp_agent`, `tlamatini_planner_trace_replay`, `tlamatini_static_version_bumper`, `todoist`, `trello`, `weather`

Assistant skill sets

- .claude/skills (canonical), .gemini/skills (mirror), .codex/skills; counted by `skill_inventory.py`

External MCPs — universal MCP client

Client

- `external_mcp_manager.py`: four transports, background connects, negative cache, supervisor; remote tools as `ext__<server>__<tool>`; 10 supervisor tools

Runtime provisioner

- `runtime_provisioner.py`: private node, npm, npx, pnpm, uv, uvx in `%LOCALAPPDATA%`; SHA-256 verified; no admin; npx shims rewritten by `win_shim.py`

Catalog and defaults

- `external_mcps.json` (user state, at most 5 active); `external_mcp_defaults.py` seeds `memory` and `sequential-thinking`, inactive

UI and triage

- External › MCPs dialog `external_mcps_dialog.js`; static triage agent `mcp_doctor.py`; runbook `adding_external_mcp`

Built-in providers and the root MCP server

Context providers

- System-Metrics: `mcp_system_server.py` (WebSocket 8765); Files-Search: `mcp_files_search_server.py` (gRPC 50051); both started by `agent/apps.py`

Root MCP server

- `tlamatini_mcp_server.py` publishes every pool agent as an MCP tool for Claude Code, Kimi and other clients (`.mcp.json`, `TLAMATINI_MCP.md`)

One definition of Windows spawning

- `win_shim.py` prefers a real `.exe`, rewrites npm/pnpm shims to `node.exe <script.js>` and avoids cmd.exe, which cuts a multi-line prompt at its first newline; shared by ACPX, the stdio MCP server and External MCPs

Messaging bridges

TeleTlamatini

- `teletlamatini.py`: long-running Telegram bot that brings authorized users into the full Multi-Turn chat

Emailer, RecMailer, Notifier

- SMTP send through Emailer, IMAP watch through RecMailer, and in-browser notifications through Notifier

Telegrammer and Whatsapper

- `telegrammer.py`: send as the bot or as your own account; `whatsapper.py`: Meta Cloud API by default, WhatsApp Web on request

Instant Messaging Doctor

- `instant_messaging_doctor.py` diagnoses tokens, contacts, templates and the 24-hour window, and is launched after a messaging failure

Devices, engines and security bridges

Firmware

- STM32er (PlatformIO + template MCP, ST-LINK preflight), ESP32er (`platformio.ini` template), Arduiner (arduino-cli, `sketch.yaml`), ESPHomer (YAML devices)

3D engines

- Blenderer speaks to the Blender MCP add-on socket (9876); Unrealer to the Unreal MCP plugin socket (55557)

Security tooling

- Kalier (MCP-Kali-Server HTTP API), Nmapper (a locally installed nmap; connect scan by default), Discoverer (ProjectDiscovery tools through a private Go toolchain); authorized targets only

Desktop control

- Mouser, Keyboarder, Windower and Shoter drive and observe the real desktop; Playwrighter drives a real browser

Frontend module map — what each page loads, in order

Parsed from the templates' {% static %} tags. Numbers give the load order; a star marks a module loaded by more than one page.

SHEET 11

Chat page · agent_page.html (96 assets)

1 Tlamatini.ico ★	38KB	49 ini.min.js	15
2 bootstrap.min.css ★	6	50 eo_mc.jpg	534KB
3 bootstrap.bundle.min.js ★	7	51 ec_mc.jpg	540KB
4 atom-one-dark.min.css ★	1	52 eo_mo.jpg	537KB
5 ui_tokens.css ★	30	53 ec_mo.jpg	543KB
6 agent_page.css	2.5k	54 canvas.html	52
7 pdf_context_progress.css	27	55 XAIHT-Tlamatini.mp4	10.8MB
8 tools_dialog.css	559	56 agent_page_state.js	495
9 skills_dialog.css	337	57 agent_page_ui.js	521
10 external_mcps_dialog.css	689	58 pdf_context_progress.js	165
11 contacts_dialog.css	538	59 agent_page_pdf.js	286
12 access_keys_wizard.css	371	60 purify.min.js	7
13 avatar.css	181	61 agent_page_markdown.js	269
14 context_gauge.css	264	62 agent_page_canvas.js	679
15 chat_dictation.css	129	63 voice_runtime.js ★	103
16 model_capacity.css	313	64 interaction_speech.js ★	67
17 model_brain_tuning.css	327	65 interaction_settings.js ★	232
18 usage_dashboard.css	114	66 editor.js	35
19 ui_shell.css ★	182	67 agent_page_editing.js	308
20 markdown_canvas.css	61	68 agent_page_dialogs.js	1.4k
21 canvas_editor.css	126	69 agent_page_context.js	188
22 interactor.css	37	70 agent_page_chat.js	2.8k
23 ui_splitters.css ★	39	71 agent_page_layout.js	469
24 xterm.css ★	218	72 agent_page_init.js	1.9k
25 terminater_dock.css ★	269	73 bridge.js	305
26 interaction_settings.css ★	38	74 terminater_dock.js ★	952
27 dialog_theme.css ★	795	75 tools_dialog.js	688
28 400.css ★	49	76 skills_dialog.js	392
29 700.css ★	49	77 external_mcps_dialog.js	545
30 jquery.min.js ★	2	78 contacts_dialog.js	384
31 jquery-ui.min.js ★	6	79 access_keys_wizard.js	406
32 ui_splitters.js ★	62	80 shared-runtime-dialogs.js ★	347
33 dialog_policy.js ★	781	81 chat_page_runtime_poller.js	151
34 file_drop_router.js ★	876	82 chat_image_paste.js	390
35 highlight.min.js	1.2k	83 context_gauge.js	441
36 python.min.js	41	84 model_capacity.js	492
37 java.min.js	38	85 model_brain_tuning.js	348
38 javascript.min.js	80	86 self_modify_switch.js	282
39 c.min.js	40	87 compact_costs.js	258
40 cpp.min.js	47	88 chat_table_contrast.js	147
41 css.min.js	31	89 checkbox_bulk_toggle.js ★	319
42 sql.min.js	17	90 release_notes_renderer.js	313
43 xml.min.js	29	91 avatar_presence.js	571
44 yaml.min.js	25	92 avatar_size.js	287
45 bash.min.js	20	93 avatar.js	437
46 typescript.min.js	95	94 mic_settings.js	197
47 kotlin.min.js	46	95 chat_dictation.js	249
48 rust.min.js	28	96 usage_dashboard.js	352

Agentic Control Panel (44 assets)

1 Tlamatini.ico ★	38KB
2 bootstrap.min.css ★	6
3 bootstrap.bundle.min.js ★	7
4 ui_tokens.css ★	30
5 agentic_control_panel.css ★	2.9k
6 flow_canvas.css ★	58
7 ui_shell.css ★	182
8 ui_splitters.css ★	39
9 xterm.css ★	218
10 terminater_dock.css ★	269
11 interaction_settings.css ★	38
12 dialog_theme.css ★	795
13 400.css ★	49
14 700.css ★	49
15 jquery.min.js ★	2
16 jquery-ui.min.js ★	6
17 ui_splitters.js ★	62
18 dialog_policy.js ★	781
19 file_drop_router.js ★	876
20 spinner.svg	18
21 canvas_item_dialog.js	1.1k
22 contextual_menus.js	730
23 flow-canvas-interactions.js ★	146
24 acp-globals.js	369
25 acp-session.js	114
26 acp-undo-manager.js	128
27 acp-agent-connectors.js	1.7k
28 shared-runtime-dialogs.js ★	347
29 acp-running-state.js	370
30 acp-control-buttons.js	1.4k
31 acp-validate.js	644
32 acp-canvas-core.js	1.7k
33 acp-flow-snapshot.js	94
34 acp-canvas-undo.js	1.2k
35 acp-file-io.js	724
36 acp-layout.js	102
37 acp-parametrizer-dialog.js	563
38 acp-connection-status.js	150
39 acp-editor-tools.js	319
40 checkbox_bulk_toggle.js ★	319
41 voice_runtime.js ★	103
42 interaction_speech.js ★	67
43 interaction_settings.js ★	232
44 terminater_dock.js ★	952

Prompt Flow Panel (27 assets)

1 Tlamatini.ico ★	38KB
2 bootstrap.min.css ★	6
3 400.css ★	49
4 700.css ★	49
5 ui_tokens.css ★	30
6 agentic_control_panel.css ★	2.9k
7 prompt_flow_panel.css	183
8 flow_canvas.css ★	58
9 ui_shell.css ★	182
10 ui_splitters.css ★	39
11 xterm.css ★	218
12 terminater_dock.css ★	269
13 interaction_settings.css ★	38
14 dialog_theme.css ★	795
15 jquery.min.js ★	2
16 jquery-ui.min.js ★	6
17 ui_splitters.js ★	62
18 dialog_policy.js ★	781
19 file_drop_router.js ★	876
20 bootstrap.bundle.min.js ★	7
21 flow-canvas-interactions.js ★	146
22 prompt-flow-panel-model.js	200
23 prompt-flow-panel.js	1.3k
24 voice_runtime.js ★	103
25 interaction_speech.js ★	67
26 interaction_settings.js ★	232
27 terminater_dock.js ★	952

Entry pages, admin, usage dialog, PDF frame (36 assets)

login:	Tlamatini.ico	38KB	
login:	bootstrap.min.css	6	
login:	bootstrap.bundle.min.js	7	
login:	atom-one-dark.min.css	1	
login:	400.css	49	
login:	700.css	49	
login:	ui_tokens.css	30	
login:	ui_shell.css	182	
login:	entry_pages.css	38	
login:	login.css	14	
login:	file_drop_router.js	876	
welcome:	Tlamatini.ico	38KB	
welcome:	bootstrap.min.css	6	
welcome:	bootstrap.bundle.min.js	7	
welcome:	atom-one-dark.min.css	1	
welcome:	400.css	49	
welcome:	700.css	49	
welcome:	ui_tokens.css	30	
welcome:	ui_shell.css	182	
welcome:	entry_pages.css	38	
welcome:	welcome.css	13	
welcome:	dialog_theme.css	795	
welcome:	dialog_policy.js	781	
welcome:	file_drop_router.js	876	
welcome:	welcome_enter_default.js	92	
admin:	Tlamatini.ico	38KB	
admin:	400.css	49	
admin:	700.css	49	
admin:	admin_theme.css	41	
admin:	file_drop_router.js	876	
pdf:	pdf_viewer.css	6.4k	
pdf:	ui_tokens.css	30	
pdf:	ui_shell.css	182	
pdf:	pdf_canvas.css	19	
pdf:	file_drop_router.js	876	
pdf:	pdf_canvas_viewer.js	213	

Data model and state — 21 Django models, 221 migrations, WAL-safe copies

Fields parsed from agent/models.py. FK arrows are written as → Target. The history chart classifies each migration by its file name.

SHEET 12

AgentMessage

- `user` ForeignKey → **User**
- `conversation_user` ForeignKey → **User**
- `message` Text
- `timestamp` DateTime

Mcp

- `idMcp` Integer
- `mcpName` Char
- `mcpDescription` Char
- `mcpContent` Text

ChatAgentRun

- `owner` ForeignKey → **User**
- `runId` Char
- `toolDescription` Char
- `templateAgentDir` Char
- `runtimeDir` Char
- `logPath` Char
- `requestText` Text
- `pid` Integer
- `status` Char
- `exitCode` Integer
- `startedAt` DateTime
- `finishedAt` DateTime

UsageDaily

- `user` ForeignKey → **User**
- `day` Date
- `model` Char
- `calls` PositiveBigInteger
- `input_tokens` PositiveBigInteger
- `output_tokens` PositiveBigInteger
- `missing_output_calls` PositiveBigInteger
- `first_seen` DateTime
- `last_seen` DateTime

LLMProgram

- `idProgram` Integer
- `programName` Char
- `programLanguage` Char
- `timestamp` DateTime
- `programContent` Text

Tool

- `idTool` Integer
- `toolName` Char
- `toolDescription` Char
- `toolContent` Text

Asset

- `idAsset` Integer
- `assetName` Char
- `assetDescription` Char
- `assetContent` Text

AcpSession

- `session_uuid` Char
- `agent_id` Char
- `user` ForeignKey → **User**
- `cwd` Char
- `state_path` Char
- `transcript_path` Char
- `started_at` DateTime
- `ended_at` DateTime
- `ok` Boolean
- `pid` Integer
- `label` Char

LLMSnippet

- `idSnippet` Integer
- `snippetName` Char
- `snippetLanguage` Char
- `timestamp` DateTime
- `snippetContent` Text

Agent

- `idAgent` Integer
- `agentName` Char
- `agentDescription` Char
- `agentContent` Text

SessionState

- `user` OneToOne → **User**
- `context_path` Char
- `context_type` Char
- `context_filename` Char
- `last_active` DateTime

SkillInvocation

- `skill_name` Char
- `user` ForeignKey → **User**
- `started_at` DateTime
- `finished_at` DateTime
- `ok` Boolean
- `iterations` Integer
- `tokens` Integer
- `args_json` Text
- `output_json` Text
- `audit_path` Char
- `failure_reason` Char
- `acp_session` ForeignKey → **AcpSession**

Prompt

- `idPrompt` Integer
- `promptName` Char
- `promptContent` Text
- `category` Char
- `hidden` Boolean
- `sort_rank` Integer

AgentProcess

- `idAgentProcess` Integer
- `agentProcessDescription` Char
- `agentProcessPid` Integer

AcpAgent

- `agent_id` Char
- `command` Char
- `description` Char
- `enabled` Boolean
- `healthy` Boolean
- `last_probe_at` DateTime
- `notes` Text

CompactState

- `active` Boolean
- `strict` Boolean
- `model` Char
- `window_tokens` Integer
- `saved_external_active` Text
- `reason` Char
- `self_modify` Boolean
- `updated_at` DateTime

Omission

- `idOmission` Integer
- `omissionName` Char
- `omissionContent` Text

ContextCache

- `query_hash` Char
- `context_blob` Text
- `timestamp` DateTime

TorrenterProfile

- `user` OneToOne → **User**
- `account` UUID

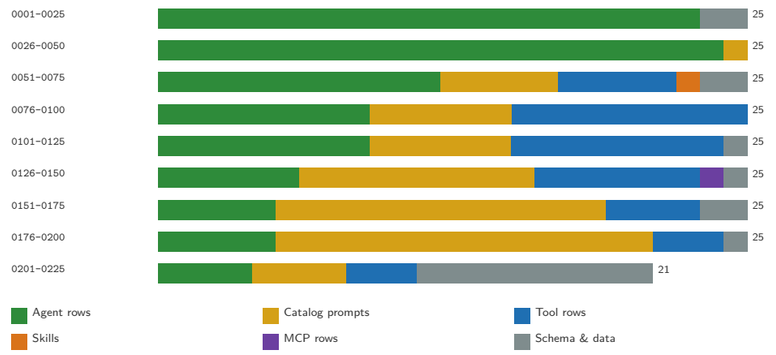
Skill

- `name` Char
- `description` Text
- `runtime` Char
- `acpx_agent` Char
- `enabled` Boolean
- `frontmatter_json` Text
- `body_sha256` Char
- `last_loaded_at` DateTime

InteractionSettings

- `user` OneToOne → **User**
- `preferences` JSON
- `revision` PositiveInteger

Migration history (25 per bar, by kind)



Backup database

- `sqlite_copy.py consistent_copy`: SQLite online backup API (reads through the WAL) → copy written in DELETE journal mode → `PRAGMA quick_check` before success

Set DB

- `move_with_sidecars` / `remove_sidecars` keep `-wal`, `-shm`, `-journal` with their own database; fail-safe, never deletes the source

Self-update carry-over

- `apply_update.ps1` copies the live DB to `DB/ToLoad` + a flag; on next launch `manage.py` swaps it back and runs `migrate` in a child (`ToLoad`, `Older`)

Chat history

- `AgentMessage` rows are the only history the model reads: `chat_history_loader.py` reloads the last 8 per request, so Drop needs no reconnect

Build, release, install and self-update pipeline

From the repository to Installer.exe, then in-app updates that keep user state; plus the guards that keep secrets out of git.

Build

Version

- `versioning.py` writes `agent/_version.py` (gitignored) from git tags; rules in `VERSIONING.md`

Inputs

- `build_runtime_assets.py` (fail-closed inventory), `hook-numpy`, `hook-torch`, `hook-pymupdf`, `requirements.txt`

Freeze

- `build.py`: PyInstaller onedir of `manage.py`, `collectstatic`, carried Python, hidden imports, `verify_frozen_agent_modules()`, gates `check_agent_runtimes.py` and `check_prompt_flow_panel.py`

Self-modify (optional)

- `-self-modify` → `copy_source_assets.py` writes `TlamatiniSourceCode/` and rebuild instructions; bundles `agent/Tlamatini.md`

Package

- `pkg.zip` → `build_uninstaller.py` (`Uninstaller.exe`) → `build_installer.py` (`Installer.exe`); `build_all.cmd` end-to-end

Release

- `build_complete_private_release.py` / `build_complete_public_release.py`; private data audited by `check_private_data.py`; keys toggled by `regen_secrets.py`

Installer.exe

Install

Setup window

- `install.py` + `setup_wizard.py` (one window for Installer and Uninstaller), transcripts by `setup_logging.py`

Recovery

- `setup_recovery.py`: verified recovery copy and journal before any change; `setup_windows.py`: snapshot of Tlamatini's own registry keys and shortcut

Windows integration

- `CreateShortcut.ps1`, `flow_file_associations.ps1` (.flw and .fpmt), `register_flw.ps1`, `register_fpmt.ps1`; `windows_app_registration.py` + `agent_manifest.py` (HKCU discovery)

Uninstall

- `uninstall.py` + `uninstall_processes.py` (owned workers only); keeps `agents/` and non-empty user folders listed in `preserved_user_state.json`

Run and update

Launch

- `Tlamatini.ps1` or `Tlamatini.exe` → `manage.py` runserver; Explorer-opened flows handled by `flow_file_open.py`

Check

- About › Check for updates → `self_update.py`: latest GitHub release, download and stage; the dialog is sealed while it downloads

Swap

- `apply_update.ps1` runs outside the install: stops the tree, `agents` → `agents_backup`, keeps the preserve set, stages the DB in `DB/ToLoad`

Migrate

- Next launch swaps the DB back and runs `migrate` in a child; daily copies by `backup_db.py`

Security and hygiene

Active defense

- `tlamatini_defender.ps1`, `tlamatini_whitelist_v2.ps1` (Controlled Folder Access), `enable_tlamatini_v2.bat`

Repository guards

- `test_private_data_guard.py`, `git_deny_go.py`, `name-guard.yml`, `finding-policy.json`

Dependencies

- `audit_dependencies.py`, `check_requirements_coverage.py`, `requirements-esphome.txt`

Secrets

- `regen_secrets.py` toggles placeholders and keys from the gitignored `data.keys`; `uncommit-keys.bat`; `private_targets.example.json`

What ships and what survives

Install layout

- `Tlamatini.exe` + `_internal/` (frozen Django app and collected statics), carried `python/`, `jre/` and `git/`, the `agents/` templates, `DB/`, `Temp/`, `Templates/`

Kept by self-update

- `config.json`, `DB`, `Temp`, `Templates`, `application(s)`, the `*_generated` and `context_files` folders and `data.keys`; the package delivers its verified uninstaller

Self-modify snapshot

- `TlamatiniSourceCode/` with a manifest and rebuild instructions; secrets redacted and heavy media omitted by `copy_source_assets.py`

Inclusion sweeps

- `sweep_self_update.py` and `sweep_self_modify.py` prove both pipelines carry every new asset

Verification and knowledge — how Tlamatini is tested and documented

208 test modules hold 6386 test functions; every automated run is visible. The documentation graph is what every assistant reads first.

Unit and contract tests

- 180 modules in `agent/` (`test_*.py`) + `acpx/tests.py`; e.g. `test_status_vocabulary.py`, `test_agent_display_names.py`, `test_dialog_dismissal_policy.py`, `test_ask_execs_allowlist.py`, `test_console_shield.py`

Visible end-to-end

- `tests_e2e`: headed Chrome against the real GUI, e.g. `test_self_healing_visual.py`, `test_create_flow_visual.py`, `test_db_backup_set_visible.py`

Daily chat harness

- `run_test.py` with `preflight.py` (repairs DB, server and login before Chrome opens), `qualify.py` and Shoter photos (`shoter_shot.py`); Gemini mirror

Visible and release checks

- `scripts/`: `*_visible.py` runners, `release_*_checks.py`, setup lifecycle cycles; JS gates `check_js_parse.mjs` + `eslint.config.mjs`; Python gate: ruff

Policy

- Headless runs are forbidden: `TestsVisiblesAndVisibleExecutionFromClaude2Codex.md`; screenshots only by the Shoter agent; tests must establish their own environment

Onboarding chain

- `CLAUDE.md` @-imports `architecture`, `multi-turn`, `exec-report`, `agents`, `acpx`, `mcp-tools`, `frontend`, `gotchas`; on demand `recent-fixes`

Other assistants

- `AGENTS.md`, `GEMINI.md`, `KIMI.md`; authoring guides `create_new_agent.md`, `create_new_mcp.md`, `create_new_skill.md`

Feature guides and records

- `docs/` feature guides (e.g. `setup-wizards`, `canvas-refresh`, `model_configuration`) and dated records in `docs/changes/` (e.g. `v1.88.0`)

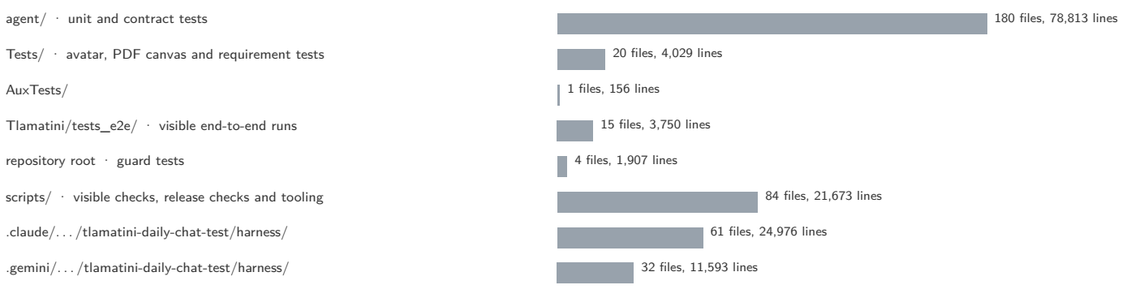
User-facing

- `README.md`, `BookOfTlamatini.md`, `agents_descriptions.md` (tooltips and canvas descriptions), `ACPX.md`, `TLAMATINI_MCP.md`

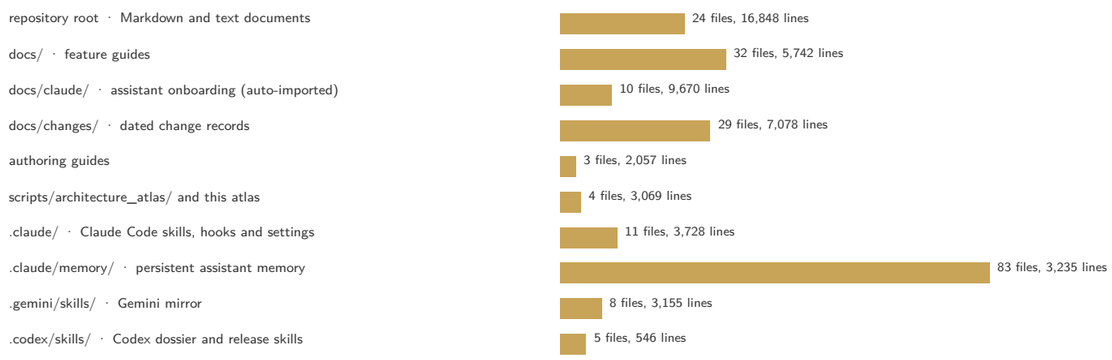
Memory, hooks and skills

- `.claude/memory/` notes, `announce_skills.py` (SessionStart), `.claude/settings.json`; four skill locations checked by `skill_inventory.py`

Verification assets per cluster



Knowledge assets per cluster



Largest test suites (test functions per file, counted with ast)

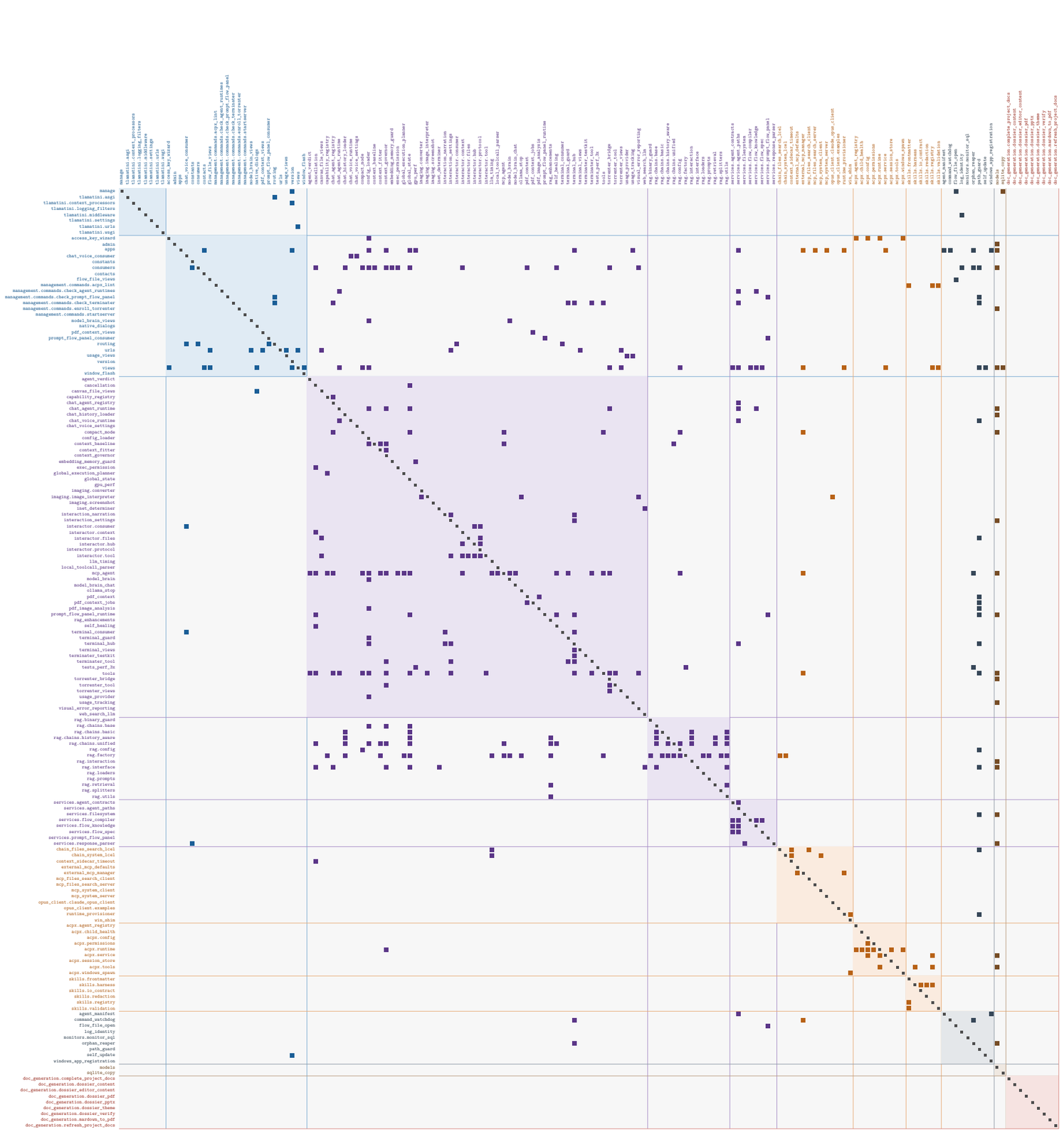




Python module dependency matrix (DSM)

Row imports column (module-level and lazy in-function imports). 160 backend modules, 382 dependencies, parsed with ast.

SHEET 15



Most imported (fan-in)

models: 21
path_guard: 17
config_loader: 16
global_state: 14
services.agent_paths: 13
context_governor: 13
terminal_hub: 12
cancellation: 10
torrenter_bridge: 7
interaction_settings: 6
chat_agent_registry: 6
rag.utils: 6

Most importing (fan-out)

views: 26
mcp_agent: 25
rag.factory: 22
tools: 19
apps: 19
consumers: 19
rag.chains.unified: 15
rag.interface: 10
urls: 9
management.commands.check_terminater: 8
prompt_flow_panel_runtime: 7
rag.chains.history_aware: 7

Import cycles (strongly connected)

- 4 modules: compact_mode, mcp_agent, rag.config, tools
- 3 modules: interaction_narration, interaction_settings, terminal_hub

Reading the matrix

- Diagonal = the module itself; shaded squares = groups (Django project, transport, intelligence, rag, services, integration, acpx, skills, platform, persistence, doc generation). Marks below the diagonal point back to groups listed earlier.
- Pool agents are excluded: they run as separate processes and import nothing from `agent.*`.

Atlas I — Server core: process, web tier and intelligence

Django/Channels process, HTTP and WebSocket entry points, the RAG chains, the Multi-Turn executor and its services.

Python · JavaScript · CSS · HTML · Markdown · YAML/JSON · PS1/BAT/CMD · Prompts/flows · C++/Arduino · Text/other · Binary/media

SHEET 16

Tlamatini/ · process entry

2 files · 1,794 lines

manage.py: log tee, console shield, colours, django_port, Temp pin, post-update DB swap

cat_art.py	64	manage.py	1.7k

Tlamatini/tlamatini/ · Django project

8 files · 749 lines

settings (SQLite WAL, WhiteNoise, temp pin), ASGI app, root URLs, per-user log-tag middleware

__init__.py	10	context_processors.py	31	middleware.py	197	urls.py	18
asgi.py	34	logging_filters.py	31	settings.py	400	wsgi.py	28

agent/ · transport and web endpoints

19 files · 18,983 lines

Channels consumers, HTTP views, app boot (AgentConfig.ready), admin, fixed chat lines

__init__.py 26	contacts.py 210	urls.py 230
access_key_wizard.py 703	flow_file_views.py 50	usage_views.py 55
admin.py 31	model_brain_views.py 133	version.py 388
apps.py 838	native_dialogs.py 504	views.py 13k
chat_voice_consumer.py 121	pdf_context_views.py 40	window_flash.py 146
constants.py 156	prompt_flow_panel_consumer.py 93	
consumers.py 2.5k	routing.py 24	

agent/templates/ · server-rendered pages

10 files · 1,712 lines

chat page, Agentic Control Panel, Prompt Flow Panel, login, welcome, usage dialog, admin skin

admin/base_site.html 15	agent/login.html 63
agent/agent_page.html 949	agent/prompt_flow_panel.html 131
agent/agentic_control_panel.html 358	agent/torrenter_sources.html 25
agent/interaction_settings.html 42	agent/usage_dialog.html 64
agent/interactor.html 23	agent/welcome.html 42

agent/management/ · manage.py commands

7 files · 704 lines

startserver plus build gates (check_agent_runtimes, check_prompt_flow_panel) and acpx_lint

__init__.py 10	commands/check_terminater.py 227
commands/acpx_lint.py 123	commands/enroll_torrenter.py 29
commands/check_agent_runtimes.py 109	commands/startserver.py 56
commands/check_prompt_flow_panel.py 150	

agent/ · intelligence core

40 files · 25,559 lines

Multi-Turn executor, tool surface, planner, self-healing, Model Brain, context fitting, prompts

agent_verdict.py 657	llm_timing.py 160
cancellation.py 199	local_toolcall_parser.py 376
capability_registry.py 691	mcp_agent.py 4.1k
chat_agent_registry.py 2.4k	model_brain.py 549
chat_agent_runtime.py 638	model_brain_chat.py 178
chat_history_loader.py 117	model_profiles.json 102
chat_voice_runtime.py 242	ollama_stop.py 68
chat_voice_settings.py 70	pdf_context.py 202
compact_mode.py 685	pdf_context_jobs.py 113
config.json 299	pdf_image_analysis.py 38
config_loader.py 135	prompt.pmt 401
context_baseline.py 489	prompt_flow_panel_runtime.py 207
context_fitter.py 760	rag_enhancements.py 687
context_governor.py 2.0k	self_healing.py 523
embedding_memory_guard.py 433	Tlamatini.md 550
exec_permission.py 259	tools.py 5.3k
global_execution_planner.py 526	usage_provider.py 333
global_state.py 53	usage_tracking.py 108
gpu_perf.py 420	visual_error_reporting.py 29
inet_determiner.py 219	web_search_llm.py 279

agent/interactor/ - live editor module

8 files · 644 lines

authenticated tab bridge, fresh buffer reads, typed teaching segments and learner handoffs

__init__.py 2	context.py 35	files.py 49	protocol.py 56
consumer.py 74	editor.mjs 154	hub.py 128	tool.py 146

agent/rag/ · retrieval-augmented generation

11 files · 3,637 lines

chain factory, ask_rag interface, FAISS + BM25 retrieval, loaders, splitters, binary guard

__init__.py 39	factory.py 1.2k	loaders.py 51	splitters.py 13
binary_guard.py 516	interaction.py 67	prompts.py 33	utils.py 191
config.py 445	interface.py 967	retrieval.py 114	

agent/rag/chains/ · answer chains

4 files · 2,075 lines

prompt-only, history-aware RAG and unified tool-agent chains

base.py 177	basic.py 193	history_aware.py 587	unified.py 1.1k

agent/services/ · flow compiler and contracts

9 files · 2,723 lines

FlowSpec, flow compiler, agent contracts and paths, response parser, prompt-flow interpreter

__init__.py 25	filesystem.py 141	flow_spec.py 174
agent_contracts.py 501	flow_compiler.py 441	prompt_flow_panel.py 438
agent_paths.py 178	flow_knowledge.py 147	response_parser.py 678

agent/imaging/ · image analysis

3 files · 681 lines

dual-backend (Claude + Qwen) interpretation, conversion, screenshots

converter.py 25	image_interpreter.py 399	screenshot.py 257

agent/opus_client/ · Claude API client

3 files · 1,396 lines

chat, images, PDFs, streaming, tools

claude_opus_client.py 969	examples.py 9	README.md 418

agent/monitors/ · SQL monitor

1 file · 83 lines

monitor_sql.py 83

agent/doc_generation/ · project dossier generator

10 files · 7,082 lines

builds and verifies the summary PDF and the PPTX deck from source facts

complete_project_docs.py 639	dossier_pptx.py 1.2k	refresh_project_docs.py 16
dossier_content.py 2.6k	dossier_theme.py 385	test_output.pdf 41KB
dossier_editor_content.py 295	dossier_verify.py 314	
dossier_pdf.py 1.5k	mardown_to_pdf.py 119	

agent/ · other modules

12 files · 2,704 lines

canvas_file_views.py 331	terminal_guard.py 483	terminater_tool.py 429
interaction_narration.py 131	terminal_hub.py 347	torrenter_bridge.py 116
interaction_settings.py 162	terminal_views.py 69	torrenter_tool.py 64
terminal_consumer.py 271	terminater_testkit.py 81	torrenter_views.py 220

Atlas II — Integration surfaces, platform and state

Built-in MCP servers, the universal external-MCP client, ACPX, the skill harness, platform guards and persistence core.

Python · JavaScript · CSS · HTML · Markdown · YAML/JSON · PS1/BAT/CMD · Prompts/flows · C++/Arduino · Text/other · Binary/media

SHEET 17

agent/ · context providers and external-MCP client

16 files · 6,914 lines

System-Metrics (WS 8765), Files-Search (gRPC 50051), sidecars, universal MCP client, runtimes, Windows shim

chain_files_search_lcel.py 791	filesearch_pb2.py 57
chain_system_lcel.py 282	filesearch_pb2_grpc.py 152
command_togen_pb2.txt 32	mcp_files_search_client.py 462
context_sidecar_timeout.py 163	mcp_files_search_server.py 240
external_mcp_defaults.py 245	mcp_system_client.py 141
external_mcp_manager.py 2.7k	mcp_system_server.py 235
external_mcps.json 4	runtime_provisioner.py 1.2k
filesearch.proto 33	win_shim.py 189

agent/acpx/ · ACPX runtime

11 files · 5,610 lines

external coding-agent CLIs as children: registry, transports, delivery verdict, 12 tools

__init__.py 135	config.py 549	service.py 112	tools.py 716
agent_registry.py 393	permissions.py 108	session_store.py 145	windows_spawn.py 180
child_health.py 299	runtime.py 1.3k	tests.py 1.6k	

agent/skills/ · skill harness

7 files · 1,829 lines

SKILL.md frontmatter, registry, planning-handoff harness, I/O contract, redaction, validator

__init__.py 29	harness.py 603	redaction.py 234	validation.py 335
frontmatter.py 186	io_contract.py 147	registry.py 295	

agent/skills_pkg/ · SKILL.md packages

39 files · 4,586 lines

29 packages loaded by boot_skills() and invoked through list_skills / invoke_skill

_meta/lint.py 85	
_meta/schema.json 101	
acp_router/SKILL.md 106	
adding_external_mcp/references/external_mcp_catalog_format.md 127	
adding_external_mcp/references/llm_reflection_research.md 112	
adding_external_mcp/references/transport_guide.md 143	
adding_external_mcp/references/troubleshooting.md 164	
adding_external_mcp/SKILL.md 297	
code_review/SKILL.md 133	
create_new_agent/SKILL.md 156	
create_new_mcp/SKILL.md 147	
flow_making/references/flw_schema.md 89	
flow_making/scripts/make_flow.py 178	
flow_making/scripts/result_to_flw.py 236	
flow_making/SKILL.md 164	
github/SKILL.md 78	
gmail/SKILL.md 59	
hello_world/SKILL.md 54	
jira/SKILL.md 65	
kali_pentest/SKILL.md 128	
notion/SKILL.md 77	
roblox_studio/SKILL.md 177	
security_audit/SKILL.md 113	
setup_new_acpx_key/SKILL.md 274	
skill_creator/scripts/quick_validate.py 63	
skill_creator/SKILL.md 129	
slack/SKILL.md 60	
summarize/SKILL.md 60	
tlamatini_allowed_hosts_tighten/SKILL.md 64	
tlamatini_csrf_exempt_audit/SKILL.md 100	
tlamatini_exec_report_row_adder/SKILL.md 85	
tlamatini_flow_from_objective/SKILL.md 126	
tlamatini_flw_doctor/SKILL.md 130	
tlamatini_new_acp_agent/SKILL.md 146	
tlamatini_planner_trace_replay/SKILL.md 85	
tlamatini_static_version_bumper/SKILL.md 103	
todoist/SKILL.md 58	
trello/SKILL.md 57	
weather/SKILL.md 57	

agent/ · platform, safety and lifecycle

8 files · 3,636 lines

Temp/Templates policy, orphan reaper, watchdog, log tags, self-update, companion discovery

agent_manifest.py 277	log_identity.py 358	self_update.py 500
command_watchdog.py 602	orphan_reaper.py 895	windows_app_registration.py 285
flow_file_open.py 312	path_guard.py 407	

persistence core

5 files · 894 lines

Django models, WAL-safe SQLite copy (online backup API), DB staging for self-update

agent/models.py 310	agent/sqlite_copy.py 407	DB/ToLoad/README.md 63
agent/run_db_wal_tests.ps1 54	DB/Older/README.md 60	

Tlamatini/jd-cli/ · bundled Java decompiler

2 files · 29 lines

jd-cli.bat 29	jd-cli.jar 3.8MB

agent/images/ · documentation media

5 files · 0 lines

ACPXKeysConfigureWizard.jpg 118KB	MenuConfig.jpg 18KB	XAIHT-Tlamatini.mp4 10.8MB
ConfigureModels.jpg 44KB	mockup.jpg 34KB	

agent/TlamatiniSourceCode/ · self-modify snapshot slot

1 file · 7 lines

filled by copy_source_assets.py in –self-modify builds

README.md 7

agent/test_fixtures/

1 file · 1 lines

recorded model output for offline tests

talker_orpheus_tokens.txt 1

Atlas IV — Presentation layer: first-party frontend

Scripts and stylesheets grouped by the page template that loads them (parsed from the templates).

Python · JavaScript · CSS · HTML · Markdown · YAML/JSON · PS1/BAT/CMD · Prompts/flows · C++/Arduino · Text/other · Binary/media

SHEET 18

Torrenter workspace

4 files · 355 lines

Signed-in queue, inspection and verified output views

layout.js 62	workspace.css 53	workspace.html 28	workspace.js 212

static/agent/interactor/ - tutor controls

2 files · 342 lines

browser bridge, local field completion and learner-first interruption

bridge.js 305	interactor.css 37

agent/static/agent/js/ · chat page (agent_page.html)

32 files · 16,786 lines

access_keys_wizard.js 406	agent_page_ui.js 521	external_mcps_dialog.js 545
agent_page_canvas.js 679	avatar.js 437	mic_settings.js 197
agent_page_chat.js 2.8k	avatar_presence.js 571	model_brain_tuning.js 348
agent_page_context.js 188	avatar_size.js 287	model_capacity.js 492
agent_page_dialogs.js 1.4k	chat_dictation.js 249	pdf_context_progress.js 165
agent_page_editing.js 308	chat_image_paste.js 390	release_notes_renderer.js 313
agent_page_init.js 1.9k	chat_page_runtime_poller.js 151	self_modify_switch.js 282
agent_page_layout.js 469	chat_table_contrast.js 147	skills_dialog.js 392
agent_page_markdown.js 269	compact_costs.js 258	tools_dialog.js 688
agent_page_pdf.js 286	contacts_dialog.js 384	usage_dashboard.js 352
agent_page_state.js 495	context_gauge.js 441	

agent/static/agent/js/ · Agentic Control Panel

17 files · 11,441 lines

acp-agent-connectors.js 1.7k	acp-file-io.js 724	acp-session.js 114
acp-canvas-core.js 1.7k	acp-flow-snapshot.js 94	acp-undo-manager.js 128
acp-canvas-undo.js 1.2k	acp-globals.js 369	acp-validate.js 644
acp-connection-status.js 150	acp-layout.js 102	canvas_item_dialog.js 1.1k
acp-control-buttons.js 1.4k	acp-parametrizer-dialog.js 563	contextual_menus.js 730
acp-editor-tools.js 319	acp-running-state.js 370	

agent/static/agent/js/ · Prompt Flow Panel

2 files · 1,469 lines

prompt-flow-panel-model.js 200	prompt-flow-panel.js 1.3k

agent/static/agent/js/ · shared by several pages

10 files · 3,885 lines

checkbox_bulk_toggle.js 319	interaction_settings.js 232	ui_splitters.js 62
dialog_policy.js 781	interaction_speech.js 67	voice_runtime.js 103
file_drop_router.js 876	shared-runtime-dialogs.js 347	
flow-canvas-interactions.js 146	terminater_dock.js 952	

agent/static/agent/js/ · entry pages, PDF frame and indirect loads

3 files · 333 lines

agentic_control_panel.js 28	pdf_canvas_viewer.js 213	welcome_enter_default.js 92

agent/static/agent/css/ · chat page (agent_page.html)

15 files · 6,545 lines

access_keys_wizard.css 371	contacts_dialog.css 538	model_capacity.css 313
agent_page.css 2.5k	context_gauge.css 264	pdf_context_progress.css 27
avatar.css 181	external_mcps_dialog.css 689	skills_dialog.css 337
canvas_editor.css 126	markdown_canvas.css 61	tools_dialog.css 559
chat_dictation.css 129	model_brain_tuning.css 327	usage_dashboard.css 114

agent/static/agent/css/ · Prompt Flow Panel

1 file · 183 lines

prompt_flow_panel.css 183

agent/static/agent/css/ · shared by several pages

8 files · 4,299 lines

agentic_control_panel.css 2.9k	interaction_settings.css 38	ui_splitters.css 39
dialog_theme.css 795	terminater_dock.css 269	ui_tokens.css 30
flow_canvas.css 58	ui_shell.css 182	

agent/static/agent/css/ · entry pages, PDF frame and indirect loads

5 files · 125 lines

admin_theme.css 41	login.css 14	welcome.css 13
entry_pages.css 38	pdf_canvas.css 19	

agent/static/ · images, avatar, PDF frame, sounds, video

10 files · 70 lines

img/avatar/ec_mc.jpg 540KB	img/Tlamatini.ico 38KB
img/avatar/ec_mo.jpg 543KB	pdf/canvas.html 52
img/avatar/eo_mc.jpg 534KB	sounds/hypervisor_alert.wav 215KB
img/avatar/eo_mo.jpg 537KB	sounds/notification.wav 279KB
img/spinner.svg 18	video/XAIHT-Tlamatini.mp4 10.8MB

Atlas V — Vendored third-party frontend and PDF.js

Pinned, offline copies served by WhiteNoise: no CDN at runtime.

Python · JavaScript · CSS · HTML · Markdown · YAML/JSON · PS1/BAT/CMD · Prompts/flows · C++/Arduino · Text/other · Binary/media

SHEET 19

vendor/frontend

1 file · 315 lines

Vendored-frontend manifest (versions + integrity)

manifest.json	315

vendor/frontend/bootstrap

5 files · 36 lines

Bootstrap 5 (navbar, dropdowns, modals, grid)

dist/css/bootstrap.min.css	6
dist/css/bootstrap.min.css.map	1
dist/js/bootstrap.bundle.min.js	7
dist/js/bootstrap.bundle.min.js.map	1
LICENSE	21

vendor/frontend/dompurify

3 files · 210 lines

DOMPurify HTML sanitizer used by the Markdown canvas

dist/purify.min.js	7
dist/purify.min.js.map	1
LICENSE	202

vendor/frontend/highlightjs

17 files · 1,795 lines

highlight.js syntax highlighter, languages and theme

highlight.min.js	1.2k
languages/bash.min.js	20
languages/c.min.js	40
languages/cpp.min.js	47
languages/css.min.js	31
languages/ini.min.js	15
languages/java.min.js	38
languages/javascript.min.js	80
languages/kotlin.min.js	46
languages/python.min.js	41
languages/rust.min.js	28
languages/sql.min.js	17
languages/typescript.min.js	95
languages/xml.min.js	29
languages/yaml.min.js	25
LICENSE	29
styles/atom-one-dark.min.css	1

vendor/frontend/jquery

3 files · 23 lines

jQuery

dist/jquery.min.js	2
dist/jquery.min.map	1
LICENSE.txt	20

vendor/frontend/jquery-ui

2 files · 49 lines

jQuery UI (dialogs, resizing)

jquery-ui.min.js	6
LICENSE.txt	43

vendor/frontend/marked

1 file · 44 lines

Marked Markdown parser (license)

LICENSE	44

vendor/frontend/nunito

23 files · 191 lines

Nunito web font (site typography)

400.css	49
700.css	49
files/nunito-cyrillic-400-normal.woff	10KB
files/nunito-cyrillic-400-normal.woff2	8KB
files/nunito-cyrillic-700-normal.woff	10KB
files/nunito-cyrillic-700-normal.woff2	8KB
files/nunito-cyrillic-ext-400-normal.woff	14KB
files/nunito-cyrillic-ext-400-normal.woff2	11KB
files/nunito-cyrillic-ext-700-normal.woff	14KB
files/nunito-cyrillic-ext-700-normal.woff2	11KB
files/nunito-latin-400-normal.woff	20KB
files/nunito-latin-400-normal.woff2	16KB
files/nunito-latin-700-normal.woff	20KB
files/nunito-latin-700-normal.woff2	16KB
files/nunito-latin-ext-400-normal.woff	21KB
files/nunito-latin-ext-400-normal.woff2	16KB
files/nunito-latin-ext-700-normal.woff	21KB
files/nunito-latin-ext-700-normal.woff2	16KB
files/nunito-vietnamese-400-normal.woff	8KB
files/nunito-vietnamese-400-normal.woff2	6KB
files/nunito-vietnamese-700-normal.woff	8KB
files/nunito-vietnamese-700-normal.woff2	6KB
LICENSE	93

vendor/frontend/xterm

2 files · 239 lines

xterm.js 5.5.0 terminal renderer of the Terminals area (Terminater)

css/xterm.css	218
LICENSE	21

vendor/frontend/xterm-addon-fit

1 file · 19 lines

xterm.js fit add-on 0.10.0 (sizes a terminal to its pane)

LICENSE	19

vendor/frontend/xterm-addon-web-links

1 file · 19 lines

xterm.js web-links add-on 0.11.0 (clickable links in a terminal)

LICENSE	19

vendor/canvas-editor - offline CodeMirror

3 files · 1,038 lines

editable text and tutoring extension; manifest and licenses carried locally

editor.js	35
LICENSES.txt	953
manifest.json	50

vendor/pdfjs

2 files · 178 lines

PDF.js license and pinned version

LICENSE	177
VERSION	1

vendor/pdfjs/build

2 files · 93,350 lines

PDF.js core library and worker

pdf.mjs	28k
pdf.worker.mjs	65k

vendor/pdfjs/cmaps

169 files · 36 lines

PDF.js binary CMaps for CJK and legacy encodings

78-EUC-H.bcmap	2KB
78-EUC-V.bcmap	173B
78-H.bcmap	2KB
78-RKSJ-H.bcmap	2KB
78-RKSJ-V.bcmap	173B
78-V.bcmap	169B
78ms-RKSJ-H.bcmap	3KB
78ms-RKSJ-V.bcmap	290B
83pv-RKSJ-H.bcmap	905B
90ms-RKSJ-H.bcmap	721B
90ms-RKSJ-V.bcmap	290B
90msp-RKSJ-H.bcmap	715B
90msp-RKSJ-V.bcmap	291B
90pv-RKSJ-H.bcmap	982B
90pv-RKSJ-V.bcmap	260B
Add-H.bcmap	2KB
Add-RKSJ-H.bcmap	2KB
Add-RKSJ-V.bcmap	287B
Add-V.bcmap	282B
Adobe-CNS1-0.bcmap	317B
Adobe-CNS1-1.bcmap	371B
Adobe-CNS1-2.bcmap	376B
Adobe-CNS1-3.bcmap	401B
Adobe-CNS1-4.bcmap	405B
Adobe-CNS1-5.bcmap	406B
Adobe-CNS1-6.bcmap	406B
Adobe-CNS1-UCS2.bcmap	40KB
Adobe-GB1-0.bcmap	217B
Adobe-GB1-1.bcmap	250B
Adobe-GB1-2.bcmap	465B
Adobe-GB1-3.bcmap	470B
Adobe-GB1-4.bcmap	601B

vendor/pdfjs/cmaps *(continued)*

169 files · 36 lines

PDF.js binary CMaps for CJK and legacy encodings

Adobe-GB1-5.bcmap	625B
Adobe-GB1-UCS2.bcmap	33KB
Adobe-Japan1-0.bcmap	225B
Adobe-Japan1-1.bcmap	226B
Adobe-Japan1-2.bcmap	233B
Adobe-Japan1-3.bcmap	242B
Adobe-Japan1-4.bcmap	337B
Adobe-Japan1-5.bcmap	430B
Adobe-Japan1-6.bcmap	485B
Adobe-Japan1-UCS2.bcmap	40KB
Adobe-Korea1-0.bcmap	241B
Adobe-Korea1-1.bcmap	386B
Adobe-Korea1-2.bcmap	391B
Adobe-Korea1-UCS2.bcmap	23KB
B5-H.bcmap	1KB
B5-V.bcmap	142B
B5pc-H.bcmap	1KB
B5pc-V.bcmap	144B
CNS-EUC-H.bcmap	2KB
CNS-EUC-V.bcmap	2KB
CNS1-H.bcmap	706B
CNS1-V.bcmap	143B
CNS2-H.bcmap	504B
CNS2-V.bcmap	93B
ETen-B5-H.bcmap	1KB
ETen-B5-V.bcmap	158B
ETenms-B5-H.bcmap	101B
ETenms-B5-V.bcmap	172B
ETHK-B5-H.bcmap	4KB
ETHK-B5-V.bcmap	158B
EUC-H.bcmap	578B
EUC-V.bcmap	170B
Ext-H.bcmap	2KB
Ext-RKSJ-H.bcmap	2KB
Ext-RKSJ-V.bcmap	218B
Ext-V.bcmap	215B
GB-EUC-H.bcmap	549B
GB-EUC-V.bcmap	179B
GB-H.bcmap	528B
GB-V.bcmap	175B
GBK-EUC-H.bcmap	14KB
GBK-EUC-V.bcmap	180B
GBK2K-H.bcmap	19KB
GBK2K-V.bcmap	219B
GBKp-EUC-H.bcmap	14KB
GBKp-EUC-V.bcmap	181B
GBpc-EUC-H.bcmap	557B
GBpc-EUC-V.bcmap	181B
GBT-EUC-H.bcmap	7KB
GBT-EUC-V.bcmap	180B
GBT-H.bcmap	7KB
GBT-V.bcmap	176B
GBTpc-EUC-H.bcmap	7KB
GBTpc-EUC-V.bcmap	182B
H.bcmap	553B
Hankaku.bcmap	132B
Hiragana.bcmap	124B
HKdla-B5-H.bcmap	3KB
HKdla-B5-V.bcmap	148B
HKdlb-B5-H.bcmap	2KB
HKdlb-B5-V.bcmap	148B
HKgccs-B5-H.bcmap	2KB
HKgccs-B5-V.bcmap	149B
HKm314-B5-H.bcmap	2KB
HKm314-B5-V.bcmap	149B
HKm471-B5-H.bcmap	2KB
HKm471-B5-V.bcmap	149B
HKscs-B5-H.bcmap	4KB
HKscs-B5-V.bcmap	159B

vendor/pdfjs/cmaps *(continued)*

169 files · 36 lines

PDF.js binary CMaps for CJK and legacy encodings

Katakana.bcmap	100B
KSC-EUC-H.bcmap	2KB
KSC-EUC-V.bcmap	164B
KSC-H.bcmap	2KB
KSC-Johab-H.bcmap	16KB
KSC-Johab-V.bcmap	166B
KSC-V.bcmap	160B
KSCms-UHC-H.bcmap	3KB
KSCms-UHC-HW-H.bcmap	3KB
KSCms-UHC-HW-V.bcmap	169B
KSCms-UHC-V.bcmap	166B
KSCpc-EUC-H.bcmap	2KB
KSCpc-EUC-V.bcmap	166B
LICENSE	36
NWP-H.bcmap	3KB
NWP-V.bcmap	252B
RKSJ-H.bcmap	534B
RKSJ-V.bcmap	170B
Roman.bcmap	96B
UniCNS-UCS2-H.bcmap	47KB
UniCNS-UCS2-V.bcmap	156B
UniCNS-UTF16-H.bcmap	49KB
UniCNS-UTF16-V.bcmap	156B
UniCNS-UTF32-H.bcmap	51KB
UniCNS-UTF32-V.bcmap	160B
UniCNS-UTF8-H.bcmap	52KB
UniCNS-UTF8-V.bcmap	157B
UniGB-UCS2-H.bcmap	42KB
UniGB-UCS2-V.bcmap	193B
UniGB-UTF16-H.bcmap	43KB
UniGB-UTF16-V.bcmap	178B
UniGB-UTF32-H.bcmap	45KB
UniGB-UTF32-V.bcmap	182B
UniGB-UTF8-H.bcmap	46KB
UniGB-UTF8-V.bcmap	181B
UniJIS-UCS2-H.bcmap	25KB
UniJIS-UCS2-HW-H.bcmap	119B
UniJIS-UCS2-HW-V.bcmap	680B
UniJIS-UCS2-V.bcmap	664B
UniJIS-UTF16-H.bcmap	39KB
UniJIS-UTF16-V.bcmap	643B
UniJIS-UTF32-H.bcmap	40KB
UniJIS-UTF32-V.bcmap	677B
UniJIS-UTF8-H.bcmap	41KB
UniJIS-UTF8-V.bcmap	678B
UniJIS2004-UTF16-H.bcmap	39KB
UniJIS2004-UTF16-V.bcmap	647B
UniJIS2004-UTF32-H.bcmap	40KB
UniJIS2004-UTF32-V.bcmap	681B
UniJIS2004-UTF8-H.bcmap	41KB
UniJIS2004-UTF8-V.bcmap	682B
UniJISPro-UCS2-HW-V.bcmap	705B
UniJISPro-UCS2-V.bcmap	689B
UniJISPro-UTF8-V.bcmap	726B
UniJISX0213-UTF32-H.bcmap	40KB
UniJISX0213-UTF32-V.bcmap	684B
UniJISX02132004-UTF32-H.bcmap	40KB
UniJISX02132004-UTF32-V.bcmap	688B
UniKS-UCS2-H.bcmap	25KB
UniKS-UCS2-V.bcmap	178B
UniKS-UTF16-H.bcmap	26KB
UniKS-UTF16-V.bcmap	164B
UniKS-UTF32-H.bcmap	26KB
UniKS-UTF32-V.bcmap	168B
UniKS-UTF8-H.bcmap	27KB
UniKS-UTF8-V.bcmap	169B
V.bcmap	166B
WP-Symbol.bcmap	179B

vendor/pdfjs/iccs

2 files · 116 lines

PDF.js ICC color profile

CGATS001Compat-v2-micro.icc	8KB
LICENSE	116

vendor/pdfjs/standard_fonts

16 files · 390 lines

PDF.js standard fonts

FoxitDingbats.pfb	29KB
FoxitFixed.pfb	17KB
FoxitFixedBold.pfb	18KB
FoxitFixedBoldItalic.pfb	19KB
FoxitFixedItalic.pfb	18KB
FoxitSerif.pfb	19KB
FoxitSerifBold.pfb	19KB
FoxitSerifBoldItalic.pfb	20KB
FoxitSerifItalic.pfb	21KB
FoxitSymbol.pfb	16KB
LiberationSans-Bold.ttf	134KB
LiberationSans-BoldItalic.ttf	132KB
LiberationSans-Italic.ttf	158KB
LiberationSans-Regular.ttf	136KB
LICENSE_FOXIT	27
LICENSE_LIBERATION	363

vendor/pdfjs/wasm

13 files · 358 lines

PDF.js WebAssembly decoders (JBIG2, OpenJPEG, QuickJS)

jbig2.wasm	102KB
jbig2_nowasm_fallback.js	15
LICENSE_JBIG2	196
LICENSE_OPENJPEG	39
LICENSE_PDFJS_JBIG2	13
LICENSE_PDFJS_OPENJPEG	22
LICENSE_PDFJS_QCMS	19
LICENSE_QCMS	21
openjpeg.wasm	246KB
openjpeg_nowasm_fallback.js	17
qcms_bg.wasm	94KB
quickjs-eval.js	16
quickjs-eval.wasm	458KB

vendor/pdfjs/web

84 files · 16,685 lines

PDF.js viewer and its toolbar icons

images/altText_add.svg	3
images/altText_disclaimer.svg	3
images/altText_done.svg	3
images/altText_spinner.svg	30
images/altText_warning.svg	3
images/annotation-check.svg	11
images/annotation-comment.svg	16
images/annotation-help.svg	26
images/annotation-insert.svg	10
images/annotation-key.svg	11
images/annotation-newparagraph.svg	11
images/annotation-noicon.svg	7
images/annotation-note.svg	42
images/annotation-paperclip.svg	6
images/annotation-paragraph.svg	16
images/annotation-pushpin.svg	7
images/checkmark.svg	5
images/comment-actionsButton.svg	3
images/comment-closeButton.svg	3
images/comment-editButton.svg	8
images/comment-popup-editButton.svg	5
images/cursor-editorFreeHighlight.svg	6
images/cursor-editorFreeText.svg	3
images/cursor-editorInk.svg	4
images/cursor-editorTextHighlight.svg	8
images/editor-toolbar-delete.svg	5
images/editor-toolbar-edit.svg	3
images/findbarButton-next.svg	3
images/findbarButton-previous.svg	3
images/gv-toolbarButton-download.svg	3
images/loading-icon.gif	2KB
images/loading.svg	1
images/messageBar_closingButton.svg	3
images/messageBar_info.svg	3
images/messageBar_warning.svg	3
images/pages_closeButton.svg	3
images/pages_selected.svg	7
images/pages_viewArrow.svg	3
images/pages_viewButton.svg	3
images/secondaryToolbarButton-documentProperties.svg	3
images/secondaryToolbarButton-firstPage.svg	3
images/secondaryToolbarButton-handTool.svg	3
images/secondaryToolbarButton-lastPage.svg	3
images/secondaryToolbarButton-rotateCcw.svg	3
images/secondaryToolbarButton-rotateCw.svg	3
images/secondaryToolbarButton-scrollHorizontal.svg	3
images/secondaryToolbarButton-scrollPage.svg	3
images/secondaryToolbarButton-scrollVertical.svg	3
images/secondaryToolbarButton-scrollWrapped.svg	3
images/secondaryToolbarButton-selectTool.svg	3
images/secondaryToolbarButton-spreadEven.svg	3
images/secondaryToolbarButton-spreadNone.svg	3
images/secondaryToolbarButton-spreadOdd.svg	3
images/signature-properties-row-check.svg	4
images/toolbarButton-bookmark.svg	3
images/toolbarButton-currentOutlineItem.svg	3
images/toolbarButton-download.svg	4
images/toolbarButton-editorFreeText.svg	5
images/toolbarButton-editorHighlight.svg	6
images/toolbarButton-editorInk.svg	4
images/toolbarButton-editorSignature.svg	6
images/toolbarButton-editorStamp.svg	8
images/toolbarButton-menuArrow.svg	3
images/toolbarButton-menuArrowNova.svg	3
images/toolbarButton-openFile.svg	3
images/toolbarButton-pageDown.svg	3
images/toolbarButton-pageUp.svg	3
images/toolbarButton-presentationMode.svg	3
images/toolbarButton-print.svg	3
images/toolbarButton-search.svg	3
images/toolbarButton-secondaryToolbarToggle.svg	3
images/toolbarButton-signaturePropertiesError.svg	7
images/toolbarButton-signaturePropertiesVerified.svg	7
images/toolbarButton-viewAttachments.svg	3
images/toolbarButton-viewLayers.svg	3
images/toolbarButton-viewOutline.svg	3
images/toolbarButton-viewsManagerToggle.svg	3
images/toolbarButton-viewThumbnail.svg	3
images/toolbarButton-zoomIn.svg	3
images/toolbarButton-zoomOut.svg	3
images/treeitem-collapsed.svg	1
images/treeitem-expanded.svg	1
pdf_viewer.css	6.4k
pdf_viewer.mjs	9.8k

Atlas VI — Agent pool: every workflow agent template

One cluster per agent, ordered by category; MT = also a Multi-Turn chat tool (chat_agent_*).

Python · JavaScript · CSS · HTML · Markdown · YAML/JSON · PS1/BAT/CMD · Prompts/flows · C++/Arduino · Text/other · Binary/media

SHEET 20

agents/ · shared helpers
2 files · 199 lines
copied beside every pool agent: model settings (@config) and visual-error reporting

- model_settings.py 166
- visual_errors.py 33

Cleaner
3 files · 354 lines
Control Agents: The post-mortem janitor.

- __init__.py 9
- cleaner.py 335
- config.yaml 10

Croner
2 files · 421 lines
Control Agents: A clock-driven trigger. Wakes up at the configured HH:MM every day and starts its target_agents. Good for nightly…

- config.yaml 7
- croner.py 414

Ender
2 files · 689 lines
Control Agents: Terminates a flow gracefully.

- config.yaml 29
- ender.py 660

Sleeper
2 files · 396 lines · MT
Control Agents: A timed pause. Waits for a configured number of milliseconds, then starts everything in target_agents. Useful for…

- config.yaml 3
- sleeper.py 393

Starter
2 files · 520 lines
Control Agents: The entry point of every flow.

- config.yaml 25
- starter.py 495

Stopper
2 files · 504 lines
Control Agents: A surgical "kill the noisy neighbor" agent.

- config.yaml 53
- stopper.py 451

Asker
2 files · 503 lines · MT
Routing Agents: An interactive A/B chooser.

- asker.py 469
- config.yaml 34

Counter
2 files · 494 lines
Routing Agents: A persistent tally with threshold-based routing.

- config.yaml 44
- counter.py 450

Forker
2 files · 698 lines
Routing Agents: An automatic A/B router.

- config.yaml 42
- forker.py 656

Raiser
2 files · 697 lines
Routing Agents: An event-driven launcher.

- config.yaml 45
- raiser.py 652

AND
2 files · 559 lines
Logic Gates: A two-input AND gate. Fires its target_agents only when BOTH source_1 AND source_2 have produced their respective…

- and.py 542
- config.yaml 17

Barrier
3 files · 793 lines
Logic Gates: An N-input barrier (a generalized AND).

- barrier.py 507
- config.yaml 8
- test_barrier.py 278

OR
2 files · 533 lines
Logic Gates: A two-input OR gate. Fires its target_agents when EITHER source_1 OR source_2 produces its pattern, whichever…

- config.yaml 16
- or.py 517

Analyzer
2 files · 573 lines
Action Agents: A deterministic static-analysis and security scanner — it uses no LLM, so its output is reproducible.

- analyzer.py 540
- config.yaml 33

Apirer
2 files · 520 lines · MT
Action Agents: An HTTP/REST API client agent.

- apirer.py 510
- config.yaml 10

Arduiner
7 files · 1,793 lines · MT
Action Agents: Tlamatini's bridge to the Arduino CLI (https://arduino.github.io/arduino-cli/) — the official all-in-one tool whose…

- arduiner.py 1.5k
- ArduinoTemplateProject/ArduinoTemplateProject.ino 40
- ArduinoTemplateProject/README.md 53
- ArduinoTemplateProject/sketch.yaml 23
- ArduinoTemplateProject/src/Heartbeat.cpp 30
- ArduinoTemplateProject/src/Heartbeat.h 35
- config.yaml 125

AudioPlayer
2 files · 913 lines · MT
Action Agents: PLAYS an audio file through a system audio OUTPUT device (speakers / audio out) via soundfile + sounddevice — the…

- audioplayer.py 839
- config.yaml 74

Blenderer
2 files · 805 lines · MT
Action Agents: Drives a Blender instance through the OFFICIAL Blender MCP add-on's TCP socket (default localhost:9876 — the add-on…

- blenderer.py 742
- config.yaml 63

Camcorder
2 files · 718 lines · MT
Action Agents: Captures from a SYSTEM CAMERA (webcam) via OpenCV — distinct from Shoter, which captures the SCREEN.

- camcorder.py 674
- config.yaml 44

Crawler
2 files · 2,205 lines · MT
Action Agents: A developer-oriented web reader with built-in LLM analysis.

- config.yaml 39
- crawler.py 2.2k

De-Compresser
2 files · 874 lines · MT
Action Agents: Deterministic short-running archive worker.

- config.yaml 26
- de_compresser.py 848

Deleter
2 files · 793 lines · MT
Action Agents: Deletes files matching one or more patterns, with exclusions, recursion, and immediate/event modes. target_path is…

- config.yaml 29
- deleter.py 764

Discoverer
2 files · 1,455 lines · MT
Action Agents: Tlamatini's bridge to the ProjectDiscovery security-tool suite (https://github.com/projectdiscovery) for recon…

- config.yaml 114
- discoverer.py 1.3k

Dockerer
2 files · 522 lines · MT
Action Agents: A Docker / docker-compose front-end.

- config.yaml 3
- dockerer.py 519

Editor
2 files · 425 lines · MT
Action Agents: Makes a SURGICAL in-place edit to a single existing text file by replacing an EXACT string (old_string) with another…

- config.yaml 16
- editor.py 409

ESP32er
8 files · 1,838 lines · MT
Action Agents: Tlamatini's bridge to PlatformIO Core (https://platformio.org) — the open-source embedded build system whose pio CLI…

- config.yaml 109
- esp32er.py 1.4k
- ESP32TemplateProject/include/README.md 16
- ESP32TemplateProject/lib/README.md 35
- ESP32TemplateProject/platformio.ini 40
- ESP32TemplateProject/README.md 114
- ESP32TemplateProject/src/main.cpp 57
- ESP32TemplateProject/test/README.md 32

ESPHomer
8 files · 1,822 lines · MT
Action Agents: Tlamatini's bridge to ESPHome (https://esphome.io) — the system that turns ESP32 / ESP8266 / RP2040 / BK72xx boards…

- config.yaml 104
- esphomer.py 1.4k
- ESPHomeTemplateProject/.gitignore 8
- ESPHomeTemplateProject/common/base.yaml 32
- ESPHomeTemplateProject/README.md 140
- ESPHomeTemplateProject/secrets.yaml.example 15
- ESPHomeTemplateProject/tlamatini-light.yaml 47
- ESPHomeTemplateProject/tlamatini-sensor.yaml 70

Executer
2 files · 1,147 lines · MT
Action Agents: Runs an arbitrary shell command and triggers its target_agents when the command finishes.

- config.yaml 40
- executer.py 1.1k

File-Creator
2 files · 485 lines · MT
Action Agents: Writes a file with arbitrary content to a specified path, then triggers downstream agents (regardless of the write…

- config.yaml 22
- file_creator.py 463

File-Extractor
2 files · 877 lines · MT
Action Agents: The "give me the raw text" sibling of File-Interpreter.

- config.yaml 10
- file_extractor.py 867

File-Interpreter
2 files · 1,084 lines · MT
Action Agents: Reads and interprets documents — DOCX, PPTX, XLSX, PDF, TXT, TeX, CSV, HTML, RTF, and more — extracting text and…

- config.yaml 46
- file_interpreter.py 1.0k

Gitter
2 files · 560 lines · MT
Action Agents: A Git operations agent. Runs clone, pull, push, commit, checkout, branch, diff, log, status, or any custom git…

- config.yaml 27
- gitter.py 533

Globber
2 files · 716 lines · MT
Action Agents: Read-only FILE discovery by glob/filename pattern under a directory (the Claude-Glob equivalent) - returns matching…

- config.yaml 14
- globber.py 702

Googler
3 files · 3,586 lines
Action Agents: Performs block- and hang-proof indexed-web search, then either extracts readable text/raw HTML from the top results…

- config.yaml 208
- googler.py 2.0k
- googler_engines.py 1.4k

Grepper
2 files · 925 lines · MT
Action Agents: Read-only regex CONTENT search across a single file or a whole directory tree (the Claude-Grep equivalent) - returns…

- config.yaml 31
- grepper.py 894

Image-Interpreter
2 files · 1,096 lines · MT
Action Agents: TRIPLE-MODEL vision analyst.

- config.yaml 156
- image_interpreter.py 940

Instant Messaging Doctor
2 files · 758 lines · MT
Action Agents: A safe first-party messaging diagnostic agent for Telegrammer and Whatsapper.

- config.yaml 52
- instant_messaging_doctor.py 706

J-Decompiler
2 files · 525 lines · MT
Action Agents: A Java decompiler. Scans a directory (or wildcard list) for .class, .jar, .war, and .ear files and uses the bundled…

- config.yaml 11
- j_decompiler.py 514

Jenkinser
2 files · 512 lines · MT
Action Agents: Triggers a job on a Jenkins server.

- config.yaml 21
- jenkinser.py 491

Kalier
2 files · 807 lines · MT
Action Agents: Tlamatini's bridge to Kali Linux offensive-security tooling, built on the MCP-Kali-Server…

- config.yaml 104
- kalier.py 703

Keyboarder
3 files · 784 lines · MT
Action Agents: Checked Windows keyboard input, bound to one HWND/PID and stopped on foreground focus loss.

- config.yaml 26
- keyboarder.py 570
- keyboarder_input.py 188

Kuberneter
2 files · 428 lines · MT
Action Agents: A kubectl front-end. Runs commands like get, apply, logs, exec against a configured namespace, with arbitrary extra…

- config.yaml 17
- kuberneter.py 411

LaTeXer
6 files · 6,469 lines · MT
Action Agents: Tlamatini's LaTeX TYPESETTING agent — the typesetting sibling of PDFer.

- config.yaml 221
- latexer_artwork.py 273
- latexer_styles.py 154
- latexer.py 5.3k
- latexer_design.py 267
- STYLES.md 242

MCP Doctor
2 files · 704 lines · MT
Action Agents: A safe External MCP diagnostic and onboarding agent.

- config.yaml 10
- mcp_doctor.py 694

Mongoxer
2 files · 214 lines · MT
Action Agents: Executes a Python block against a MongoDB database using pymongo.

- config.yaml 18
- mongoxer.py 196

Mouser
3 files · 1,048 lines · MT
Action Agents: Mouse movement, clicks, drags, scrolling, window anchors, and reference-image matching.

- config.yaml 88
- mouser.py 783
- mouser_coordinates.py 177

Mover
2 files · 884 lines · MT
Action Agents: Moves or copies files with wildcard sources, exclusions, and optional recursion.

- config.yaml 14
- mover.py 870

NetSpeed-Calculator
2 files · 1,955 lines · MT
Action Agents: Measures THIS machine's real Internet connection and reports the answer with its error bar — download, upload…

- config.yaml 83
- netspeed_calculator.py 1.9k

Nmapper
2 files · 1,281 lines · MT
Action Agents: Tlamatini's LOCAL, USE-ONLY nmap bridge for pentesters / CTF recon.

- config.yaml 88
- nmapper.py 1.2k

PDFer
15 files · 10,824 lines · MT
Action Agents: Document composer for an answer, Markdown, HTML, plain text, images or existing PDFs.

- config.yaml 217
- pdfer_color.py 896
- pdfer_styles.py 104
- pdfer.py 1.9k
- pdfer_consult.py 378
- pdfer_tables.py 664
- pdfer_artwork.py 261
- pdfer_docmodel.py 777
- pdfer_theme.py 1.2k
- pdfer_atelier.py 1.0k
- pdfer_nuance.py 1.2k
- pdfer_typography.py 709
- pdfer_audit.py 494
- pdfer_ornament.py 702
- STYLES.md 239

Playwrighter
2 files · 1,111 lines · MT
Action Agents: A scripted, interactive browser-automation agent built on Playwright (Chromium/Firefox/WebKit).

- config.yaml 80
- playwrighter.py 1.0k

PPTXer
15 files · 11,401 lines · MT
Action Agents: Builds editable PowerPoint presentations from an answer, Markdown, outlines, or JSON, with embedded local/remote…

- config.yaml 179
- pptxer_diagram.py 433
- pptxer_media.py 693
- pptxer.py 1.2k
- pptxer_docmodel.py 731
- pptxer_nuance.py 887
- pptxer_audit.py 645
- pptxer_draw.py 866
- pptxer_render.py 944
- pptxer_build.py 1.5k
- pptxer_fonts.py 1.0k
- pptxer_theme.py 621
- pptxer_color.py 736
- pptxer_layout.py 853
- STYLES.md 100

Prompter
2 files · 473 lines · MT
Action Agents: Sends a configured prompt to an Ollama LLM and logs the response into a structured INI_SECTION_PROMPTER block.

- config.yaml 30
- prompter.py 443

PSer
2 files · 604 lines · MT
Action Agents: An LLM-powered process finder.

- config.yaml 32
- pser.py 572

Pythonxer
2 files · 875 lines · MT
Action Agents: Runs an inline Python script (provided as a YAML string) behind a STRICT correctness gate: it first compile()-parses…

- config.yaml 46
- pythonxer.py 829

Recorder
2 files · 1,041 lines · MT
Action Agents: Records AUDIO from a system input device (MICROPHONE) via sounddevice and saves a WAV (written with the stdlib wave…

- config.yaml 57
- recorder.py 984

Reviewer
2 files · 626 lines
Action Agents: An LLM-powered code reviewer.

- config.yaml 38
- reviewer.py 588

SCPer
2 files · 529 lines · MT
Action Agents: Transfers a file to or from a remote host over SCP using a pre-configured SSH key.

- config.yaml 27
- scper.py 502

Shoter
2 files · 606 lines · MT
Action Agents: Captures the whole desktop by default (all_screens: true) and saves a PNG silently.

- config.yaml 21
- shoter.py 585

SQLer
2 files · 237 lines · MT
Action Agents: Executes a Python block against a Microsoft SQL Server instance using pyodbc.

- config.yaml 20
- sqler.py 217

SSHer
2 files · 470 lines · MT
Action Agents: Runs a shell command on a remote host over SSH using a pre-configured key.

- config.yaml 22
- ssher.py 448

STM32er
2 files · 2,936 lines · MT
Action Agents: Tlamatini's STM32 firmware agent — it programs the WHOLE ST 32-bit line, from the Blue Pill (STM32F103) up through…

- config.yaml 242
- stm32er.py 2.7k

Summarizer
2 files · 670 lines · MT
Action Agents: Two modes in one agent. Polling mode: continuously watches the logs of its source_agents and feeds new content to an…

- config.yaml 57
- summarizer.py 613

Talker
2 files · 1,629 lines · MT
Action Agents: TEXT-TO-SPEECH (TTS): SPEAKS text aloud through a system audio OUTPUT device (speakers) by driving an OLLAMA…

- config.yaml 151
- talker.py 1.5k

Terminater
5 files · 4,463 lines · MT
Action Agents: Raises REAL cmd.exe / PowerShell terminals — visible terminals inside Tlamatini's Terminals area (Views ▷ Terminals;…

- config.yaml 86
- terminater_conpty.py 667
- terminater_text.py 1.1k
- terminater.py 726
- terminater_engine.py 1.9k

Torrenter
6 files · 1,087 lines · MT
Action Agents: Controls persistent Tlachihuani Jobs downloads for the signed-in account.

- activity.py 126
- config.yaml 9
- sources.py 85
- client.py 260
- contract.py 280
- torrenter.py 327

Unrealer
2 files · 1,033 lines · MT
Action Agents: Drives an Unreal Engine 5 editor instance through the Unreal MCP plugin's TCP protocol (default 127.0.0.1:55557 —…

- config.yaml 160
- unrealer.py 873

Video-Analyzer
5 files · 3,072 lines · MT
Action Agents: Video-Analyzer supports analysis_type: robotics (default), transcription, and summary.

- config.yaml 202
- video_analyzer.py 1.2k
- video_fetch.py 855
- README.md 241
- video_content.py 589

VideoPlayer
2 files · 1,043 lines · MT
Action Agents: PLAYS a video file (WITH audio) on a chosen DISPLAY (screen) — the on-screen sibling of AudioPlayer (speakers).

- config.yaml 69
- videoplayer.py 974

Atlas VI — Agent pool: every workflow agent template (continued 2)

One cluster per agent, ordered by category; MT = also a Multi-Turn chat tool (chat_agent_*).

Python · JavaScript · CSS · HTML · Markdown · YAML/JSON · PS1/BAT/CMD · Prompts/flows · C++/Arduino · Text/other · Binary/media

Whisperer
4 files · 2,230 lines · MT

Action Agents: Direct chat microphone: Mic beside Send starts host capture without an LLM decision and stops on the configured…

chat_worker.py	232	README.md	82
config.yaml	182	whisperer.py	1.7k

Windower
2 files · 963 lines · MT

Action Agents: The window manager of the desktop-UI trio (Windower = the window itself, Mouser = clicks inside it, Keyboarder =…

config.yaml	77	windower.py	886

Emailer
2 files · 836 lines · MT

Notification Agents: Sends an SMTP email when triggered.

config.yaml	147	emailer.py	689

FlowHypervisor
4 files · 1,920 lines

Notification Agents: Watches a running flow using process state, logs and current agent contracts.

config.yaml	18	hypervisor_alert.wav	215KB
flowhypervisor.py	786	monitoring-prompt.pmt	1.1k

Monitor Netstat
2 files · 340 lines · MT

Notification Agents: The network-port version of Monitor-Log.

config.yaml	42	monitor_netstat.py	298

Monitor-Log
2 files · 462 lines · MT

Notification Agents: A pure log-file watcher with LLM event detection.

config.yaml	44	monitor_log.py	418

Notifier
2 files · 509 lines · MT

Notification Agents: The browser-notification agent.

config.yaml	17	notifier.py	492

RecMailer
2 files · 430 lines · MT

Notification Agents: An IMAP email receiver with LLM analysis.

config.yaml	63	recmailer.py	367

Telegrammer
7 files · 1,409 lines · MT

Notification Agents: The single Telegram send/receive agent using official Telegram surfaces only.

config.yaml	27
guide_assets/telegram_01_botfather_search.jpg	72KB
guide_assets/telegram_02_newbot_token.jpg	124KB
guide_assets/telegram_03_start_bot_getupdates.jpg	116KB
guide_assets/telegram_04_tlamatini_contacts.jpg	119KB
HOW_TO_GET_YOUR_TELEGRAM_ASSETS.md	303
telegrammer.py	1.1k

Whatsapper
8 files · 1,437 lines · MT

Notification Agents: The single WhatsApp send/receive agent with two explicit outbound identities.

config.yaml	62
guide_assets/whatsapp_01_meta_create_app.jpg	90KB
guide_assets/whatsapp_02_api_setup_keys.jpg	204KB
guide_assets/whatsapp_03_test_recipient_send.jpg	249KB
guide_assets/whatsapp_04_system_user_token.jpg	139KB
guide_assets/whatsapp_05_tlamatini_contacts.jpg	104KB
HOW_TO_GET_YOUR_WHATSAPP_ASSETS.md	343
whatsapper.py	1.0k

FlowBacker
2 files · 478 lines

Utility Agents: A passive backup utility.

config.yaml	10	flowbacker.py	468

FlowCreator
6 files · 8,828 lines · MT

Utility Agents: Designs a workflow from your objective.

agentic_skill.md	2.6k	flow_catalog.json	5.0k	flowcreator.py	609
config.yaml	40	flow_knowledge.py	295	result_to_flw.py	236

Parametrizer
2 files · 1,258 lines

Utility Agents: Carries structured results from one agent into another agent's configuration, one segment at a time.

config.yaml	11	parametrizer.py	1.2k

Kyber-Cipher
2 files · 594 lines · MT

Cryptography Agents: Encrypts a plaintext buffer with a CRYSTALS-Kyber public key.

config.yaml	14	kyber_cipher.py	580

Kyber-DeCipher
2 files · 539 lines · MT

Cryptography Agents: The inverse of Kyber-Cipher.

config.yaml	20	kyber_decipher.py	519

Kyber-KeyGen
2 files · 491 lines · MT

Cryptography Agents: Generates a CRYSTALS-Kyber post-quantum public/private key pair in base64.

config.yaml	8	kyber_keygen.py	483

TeleTlamatini
2 files · 1,364 lines

Multi-Channel Bridges: A long-running Telegram bot that bridges authorized Telegram users into the full Multi-Turn + Exec Report Tlamatini…

config.yaml	101	teletlamatini.py	1.3k

ACPXer
2 files · 957 lines

External Coding-Agent Driver: Drives one external coding-agent CLI session per run — Claude Code, Codex, Cursor, Gemini, Qwen, the Tlamatini…

acpxer.py	914	config.yaml	43

Atlas VII — Verification: unit, contract, visible and release tests

Every automated test runs headed and visible on the real desktop; screenshots are taken by Shoter.

Python · JavaScript · CSS · HTML · Markdown · YAML/JSON · PS1/BAT/CMD · Prompts/flows · C++/Arduino · Text/other · Binary/media

SHEET 22

agent/ · unit and contract tests

180 files · 78,813 lines

180 modules, Django test runner; pin the contracts described in CLAUDE.md

test_access_key_wizard.py	145	test_crawler_agent.py	342
test_agent_display_names.py	287	test_crawler_resilience.py	503
test_agent_manifest.py	449	test_createsuperuser_wizard.py	71
test_agent_verdict.py	293	test_ctrl_c_shutdown.py	216
test_agents_see_hidden_dirs.py	71	test_db_backup_restore_wal.py	432
test_arduiner_agent.py	717	test_db_backup_safety.py	130
test_ask_execs_allowlist.py	230	test_de_compresser.py	843
test_audioplayer_agent.py	780	test_deep_research_prompt.py	86
test_binary_guard.py	445	test_deleter_safety.py	128
test_blenderer_agent.py	577	test_desktop_input_agents.py	366
test_build_env.py	151	test_dialog_dismissal_policy.py	567
test_build_pip_quiet.py	206	test_dialog_theme.py	767
test_build_root_assets.py	229	test_discoverer_agent.py	533
test_build_scripts.py	720	test_discoverer_thousand.py	141
test_camcorder_agent.py	467	test_django_port_config.py	254
test_cancellation.py	215	test_drag_divider_look.py	70
test_canvas_file_views.py	348	test_drop_message.py	331
test_catalog_pruning.py	49	test_editor_agent.py	181
test_chain_readiness.py	327	test_embedding_memory_guard.py	638
test_chat_agent_run_wait.py	200	test_esp32er_agent.py	767
test_chat_bridge_bots.py	234	test_esphomer_agent.py	554
test_chat_history_window.py	94	test_exec_report_verdict.py	142
test_chat_voice.py	492	test_execute_command_bounded.py	176
test_checkbox_bulk_toggle.py	189	test_executer_shell_request.py	34
test_command_watchdog.py	367	test_executer_window_report.py	148
test_compact_mode_pingpong.py	117	test_external_mcp_add_flow.py	179
test_compact_mode_switch.py	468	test_external_mcp_e2e.py	470
test_configure_dialog_initial_ticks.py	54	test_external_mcp_transports.py	294
test_console_colors.py	517	test_external_mcp_universal.py	674
test_console_shield.py	602	test_file_creator_append.py	241
test_contacts.py	153	test_file_drop_router.py	251
test_context_governor.py	572	test_file_extractor_reader.py	104
test_context_real_tokens.py	691	test_flow_contracts.py	227
test_context_restore_spinner.py	204	test_flow_file_open.py	160
test_context_sidecar_timeout.py	258		

agent/ · unit and contract tests *(continued)*

180 files · 78,813 lines

180 modules, Django test runner; pin the contracts described in CLAUDE.md

test_flow_file_views.py	100	test_interactor.py	319
test_flow_knowledge.py	369	test_j_decompiler_agent.py	252
test_frontend_mutable_state.py	260	test_js_gates.py	60
test_gitter_custom_command.py	228	test_kalier_agent.py	1.1k
test_glob_patterns.py	508	test_latexer_agent.py	1.3k
test_globber_agent.py	156	test_latexer_delivery_truth.py	66
test_googler_agent.py	502	test_latexer_layout_truth.py	60
test_googler_dorks.py	513	test_latexer_major_removal_and_window_repair.py	464
test_googler_resilience.py	800	test_latexer_model_repair_truth.py	55
test_grepper_agent.py	172	test_latexer_repair_ladder.py	539
test_grepper_encodings.py	222	test_latexer_styles.py	186
test_grepper_lines_mode.py	279	test_latexer_suite.py	1.7k
test_image_interpreter_agent.py	401	test_latexer_verbatim_channel.py	326
test_instant_messaging_doctor.py	171	test_log_identity.py	349
test_interaction_settings.py	196	test_mcp_configured_endpoints.py	217

agent/ · unit and contract tests *(continued)*

180 files · 78,813 lines

180 modules, Django test runner; pin the contracts described in CLAUDE.md

test_model_brain.py	511	test_sqlite_wal_mode.py	52
test_model_capacity.py	456	test_status_vocabulary.py	570
test_model_settings.py	221	test_step_by_step_mode.py	209
test_mover_file_safety.py	218	test_stm32er_agent.py	1.8k
test_native_dialogs.py	294	test_talker_agent.py	1.3k
test_netspeed_calculator_agent.py	1.6k	test_telegrammer_identity.py	214
test_nmapper_agent.py	268	test_temp_dir_policy.py	479
test_no_audio_in_tests.py	60	test_terminater_agent.py	700
test_ollama_catalog_from_server.py	282	test_terminater_engine.py	1.7k
test_ollama_stop_limit.py	250	test_terminater_hub_api.py	846
test_orphan_reaper.py	340	test_terminater_integration.py	727
test_parametrizer_mcp_doctor.py	145	test_terminater_pfp.py	807
test_password_quoting.py	605	test_terminater_regressions.py	269
test_pdf_context.py	307	test_terminater_reliability.py	1.4k
test_pdfer_agent.py	1.2k	test_terminater_text.py	583
test_pdfer_nuance_layout.py	654	test_tool_failure_recovery.py	87
test_pdfer_styles.py	192	test_torrenter.py	364
test_playwrighter_agent.py	662	test_torrenter_delivery.py	82
test_pptxer_agent.py	1.3k	test_uninstall_processes.py	129
test_pptxer_styles.py	113	test_uninstaller_lifecycle.py	295
test_pptxer_visibility.py	308	test_uninstaller_mechanics.py	672
test_preserved_user_state.py	201	test_unrealer_agent.py	518
test_program_name_collisions.py	187	test_usage.py	518
test_prompt_catalog_contiguous.py	311	test_version_guard.py	62
test_prompt_flow_panel.py	292	test_video_analyzer_agent.py	357
test_prompt_flow_panel_carriage.py	96	test_video_analyzer_content.py	459
test_prompt_flow_panel_runtime.py	198	test_video_analyzer_fetch.py	595
test_prompt_flow_panel_websocket.py	173	test_videoplayer_agent.py	758
test_prompt_shape_sentences.py	67	test_visual_fatal_errors.py	246
test_public_release_targets.py	486	test_voice_commands_catalog.py	266
test_pyinstaller_guard.py	149	test_watchdog_foreground_exemption.py	672
test_recorder_agent.py	617	test_web_process_stays_lean.py	232
test_release_payload_ownership.py	173	test_whatsapper_identity.py	202
test_runtime_provisioner.py	498	test_whisperer_agent.py	1.1k
test_security_assets_carriage.py	140	test_win_shim.py	404
test_self_healing.py	259	test_windower_agent.py	662
test_self_knowledge_is_current.py	112	test_windows_app_registration.py	107
test_self_modify_gate.py	459	test_wizard_intent_guards.py	219
test_self_modify_switch.py	481	tests.py	7.8k
test_self_update.py	157	tests_perf_3x.py	432
test_shoter_takes_screenshots.py	72		

Tests/ · avatar, PDF canvas and requirement tests

20 files · 4,029 lines

avatar_dev_server/dev_settings.py	29	test_canvas_text_decoding.cjs	87
avatar_dev_server/development_test.py	51	test_pdf_canvas_assets.py	68
run_avatar_tests.py	129	test_pdf_canvas_browser.py	738
run_presence_tests.ps1	76	test_perf_3x_visual.py	216
test_about_window_name_visible.py	66	test_release_regeneration.py	270
test_avatar_animation.cjs	196	test_release_type_banner.py	143
test_avatar_collectstatic.py	66	test_release_version_pinning.py	164
test_avatar_presence_visible.py	582	test_requirements_coverage.py	103
test_avatar_visible.cjs	170	test_setup_release_paths.py	627
test_build_env_concurrency.py	160	test_update_client_carriage.py	88

AuxTests/

1 file · 156 lines

mypdf2.py	156

Tlamatini/tests_e2e/ · visible end-to-end runs

15 files · 3,750 lines

headed Chrome against the real chat GUI

oob_child_helper.py	47	test_self_healing_visual.py	316
README.md	79	test_voice_visual.py	322
test_create_flow_visual.py	477	Tlamatini-Whatsapper-Daily-Tests.py	563
test_ctrl_c_quits_visible.py	188	torrenter_books.py	151
test_db_backup_set_visible.py	416	torrenter_console_windows.py	90
test_oob_browser_visible.py	138	torrenter_live.py	158
test_oob_namu_unit.py	190	torrenter_panel.py	222
test_prompts_catalog_visual.py	393		

repository root · guard tests

4 files · 1,907 lines

private-data guard, author banner, performance

test_author_banner.py	98	test_perf_3x_visual.py	479
test_check_private_data.py	1.0k	test_private_data_guard.py	287

scripts/ · visible checks, release checks and tooling

84 files · 21,673 lines

**_visible.py, release_*_checks.py, setup cycles, vendoring, catalog and inventory tools*

acp_editor_visible.py	291	prompt_flow_selection_visible.py	201
agent_catalog_visible.py	203	refresh_supporting_pdfs.py	60
build_canvas_editor.mjs	63	release_acp_editor_checks.py	186
build_setup_full_payload.py	70	release_acp_runtime_checks.py	79
build_setup_validation.py	58	release_agent_catalog_checks.py	146
canvas_editing_visible.py	1.2k	release_auth_checks.py	102
canvas_editor_entry.mjs	248	release_commentary_checks.py	401
canvas_refresh_visible.py	556	release_extended_checks.py	295
canvas_text_decoding_visible.py	226	release_local_agent_checks.py	114
chat_composer_resize_checks.py	264	release_mcp_checks.py	81
chat_composer_resize_visible.py	158	release_output_resize_checks.py	230
chat_composer_visible_window.py	83	release_prompt_runtime_checks.py	143
chat_microphone_visible.py	365	release_shell_file_checks.py	73
chat_viewport_visible_checks.py	156	review_documentation_visuals.py	84
chat_views_frozen_visible.py	254	run_menu_state_checks.py	290
chat_views_visible.py	480	setup_crash_agents_visible.py	274
check_js_parse.mjs	68	setup_desktop_cycles.py	138
check_python_quality.py	104	setup_lifecycle_checks.py	390
check_requirements_coverage.py	283	setup_release_cycles.py	461
docked_avatar_resize_checks.py	178	setup_wizard_visible.py	274
dossier_visual_walkthrough_visible.py	231	skill_inventory.py	176
drag_divider_look_visible.py	467	sync_assistant_skills.py	143
file_drop_router_cases.mjs	455	terminater_visible_acp.py	461
flow_canvas_mechanics_visible.py	398	terminater_visible_all.py	47
flow_file_associations_test.ps1	120	terminater_visible_battery.py	733
flow_files_visible.py	502	terminater_visible_chat.py	228
installed_panels_visible.py	575	terminater_visible_common.py	556
interaction_settings_visible.py	354	terminater_visible_pfp.py	402
interactor_runtime_server.py	106	tlamatini_clean_install.ps1	106
interactor_runtime_visible.py	391	ui_sweep_resize_checks.py	69
interactor_visible.py	764	ui_sweep_visible.py	138
markdown_canvas_visible.py	573	update_flow_catalog.py	100
menu_browser_checks.py	525	usage_dashboard_browser_checks.py	83
menu_live_checks.py	145	usage_dashboard_fixture_checks.py	383
menu_reference_graph.mjs	95	usage_dashboard_visible.py	172
panel_search_title_visible.py	367	vendor_frontend.py	95
playwright_real_visibility.cjs	21	vendor_pdfjs.py	57
pptxer_visibility_gallery.py	95	verify_latexer_styles.py	246
prompt_flow_commentary_visible.py	175	verify_pdfer_styles.py	151
prompt_flow_connections_visible.py	540	verify_pptxer_layout.py	164
prompt_flow_extension_visible.py	233	verify_pptxer_styles.py	133
prompt_flow_output_resize_visible.py	381	visual_fatal_dialog_visible.py	191

.claude/.../tlamatini-daily-chat-test/harness/

61 files · 24,976 lines

daily chat regression + feature labs (Playwright headed, Shoter screenshots)

_peek.py	27	grepper_lines_visible.py	235
_stage.ps1	75	grepper_login_probe.py	62
acp_backend_banner_test.py	358	latexer_openmp_e2e.py	446
acp_backend_banner_test.py.antes_shoter	362	latexer_stepbystep_5.py	861
acp_dialog_full_sweep.py	377	mcp_playwright_suite.py	218
acp_dialog_theme_visible.py	269	model_brain_visible.py	557
ask_execs_regression.py	676	monitor.py	178
ask_execs_regression.py.antes_shoter	679	ollama_remote_catalog_visible.py	505
bulk_toggle_behaviour.py	291	pdf_context_visible.py	424
bulk_toggle_visible.py	199	pdfer_nuance_visible.py	529
cancel_regression.py	571	playwrighter_run.py	164
cancel_regression.py.antes_shoter	575	pptxer_visible.py	781
compact_mode_visible.py	483	preflight.py	334
compact_pingpong_visible.py	342	qualify.py	188
compact_switch_visible.py	691	questions.py	403
config.py	132	README.md	118
context_gauge_100.py	500	requirements.txt	7
context_gauge_real_lab.py	508	run_test.py	726
context_gauge_visible.py	288	self_modify_visible.py	481
context_restore_spinner_visible.py	726	shoter_shot.py	100
cyber_watchdog_test.py	463	step_by_step_catalog.py	381
deepseek_remote_chat_visible.py	616	step_by_step_catalog.py.antes_shoter	381
dialog_policy_visible.py	493	stop_sequence_visible.py	582
dialog_theme_visible.py	256	talk_test.py	397
discoverer_1000.py	27	uninstaller_visible.py	466
discoverer_1000.py.antes_shoter	454	video_link_visible.py	369
drop_message_visible.py	670	voice_commands_visible.py	420
file_drop_visible.py	615	window-test-screen.py	1.5k
flowcreator_wizard_e2e.py	374	windower_focus.py	164
full_mode_visible.py	252	wrapped_questions.py	323
googler_dork_hunt.py	324		

.gemini/.../tlamatini-daily-chat-test/harness/

32 files · 11,593 lines

Gemini mirror of the daily chat harness

acp_backend_banner_test.py	358	flowcreator_wizard_e2e.py	374
acp_backend_banner_test.py.antes_shoter	362	googler_dork_hunt.py	324
acp_dialog_full_sweep.py	377	latexer_openmp_e2e.py	446
acp_dialog_theme_visible.py	269	latexer_stepbystep_5.py	861
ask_execs_regression.py	676	mcp_playwright_suite.py	218
ask_execs_regression.py.antes_shoter	679	playwrighter_run.py	164
bulk_toggle_behaviour.py	291	qualify.py	188
bulk_toggle_visible.py	199	questions.py	403
cancel_regression.py	571	README.md	118
cancel_regression.py.antes_shoter	575	requirements.txt	7
config.py	132	run_test.py	726
cyber_watchdog_test.py	463	shoter_shot.py	100
dialog_policy_visible.py	493	step_by_step_catalog.py	381
dialog_theme_visible.py	256	step_by_step_catalog.py.antes_shoter	381
discoverer_1000.py	27	talk_test.py	397
discoverer_1000.py.antes_shoter	454	wrapped_questions.py	323

Atlas VIII — Build, release, install, security and repository root

From source to Installer.exe, self-update and uninstall; guards that keep private data out of git.

Python · JavaScript · CSS · HTML · Markdown · YAML/JSON · PS1/BAT/CMD · Prompts/flows · C++/Arduino · Text/other · Binary/media

SHEET 23

build and release tooling

25 files · 11,445 lines

PyInstaller build, inventories, versioning, release wrappers, dependency and private-data audits

.gitattributes	14	git_deny_go.py	154
.gitignore	462	package.json	52
audit_dependencies.py	185	pnpm-lock.yaml	1.3k
build.py	2.4k	private_targets.example.json	43
build_all.cmd	101	pyinstaller_hooks/hook-numpy.py	158
build_complete_private_release.py	546	pyinstaller_hooks/hook-pymupdf.py	14
build_complete_public_release.py	956	pyinstaller_hooks/hook-torch.py	30
build_installer.py	564	regen_secrets.py	567
build_runtime_assets.py	524	requirements-esphome.txt	7
build_uninstaller.py	371	requirements.txt	304
check_private_data.py	721	uncommit-keys.bat	243
copy_source_assets.py	984	versioning.py	233
eslint.config.mjs	537		

install, update and launch

21 files · 5,813 lines

setup window, recovery, Windows integration, self-update swap, uninstall, launchers

apply_update.ps1	389	ollama_credits.py	387	setup_windows.py	271
backup_db.py	206	preserved_user_state.json	28	setup_wizard.py	529
CreateShortcut.json	3	register_flw.ps1	15	Tlamatini.ps1	69
CreateShortcut.ps1	264	register_fpmt.ps1	12	uninstall.py	1.6k
flow_file_associations.ps1	181	RemoveShortcut.ps1	59	uninstall_processes.py	147
freeingport8000.ps1	65	setup_logging.py	100	unregister_flw.ps1	15
install.py	880	setup_recovery.py	587	unregister_fpmt.ps1	12

root stdio MCP server

3 files · 1,764 lines

exposes every pool agent and ACPX to external MCP clients

.mcp.json	8	tlamatini_acpx.py	986	tlamatini_mcp_server.py	770

security/ · security arsenal

11 files · 2,763 lines

Active Defender, whitelist v2 (Controlled Folder Access), access tests

automated_tests_of_security_assets.py	128	test_windows_access.ps1	119
enable_tlamatini_v2.bat	734	tlamatini_defender.ps1	718
README.md	139	tlamatini_whitelist_v2.ps1	567
run_defender.bat	20	verify_agent_execution.py	132
sync_enable_launcher.py	65	windows_access_helpers.ps1	128
test_security_syntax.ps1	13		

CI and scanning policy

2 files · 193 lines

.github/workflows/name-guard.yml	99	.scanning/finding-policy.json	94

demo flows

1 file · 434 lines

ready-made .flw files for the Agentic Control Panel

MCPMemoriesFlowCreation.flw	434

repository root · media, documents and license

11 files · 555 lines

build_env.py	323		
ConfigureModels.png	146KB		
LICENSE	21		
OllamaPricing.png	178KB		
OOB_shift_reaper-and-NAMU-Design.pdf	46KB		
PromptDesigner.jpg	106KB		
pyinstaller_guard.py	211		
Tlamatini.ico	38KB		
Tlamatini.jpg	15KB		
tlamatini_app_summary.pdf	2.8MB		
Tlamatini_eXtended_Artificial_Intelligence_Humanly_Tempered.pptx	5.0MB		

Atlas IX — Knowledge and governance

Documentation, change records, assistant onboarding, memory and the four tracked skill locations.

Python · JavaScript · CSS · HTML · Markdown · YAML/JSON · PS1/BAT/CMD · Prompts/flows · C++/Arduino · Text/other · Binary/media

SHEET 24

repository root · Markdown and text documents

24 files · 16,848 lines

README, onboarding files for each assistant, design notes and session records

_acpx_pipeline_demo_report.md	65
ACPX.md	910
AGENTS.md	493
agents_descriptions.md	265
AvatarImprovementByClaude.md	416
BookOfTlamatini.md	4.7k
Claude-Desktop-KALI-MCP-Session.md	694
CLAUDE.md	1.4k
Codex2ClaudeRecommendations.md	624
DesignOfIncludingMemoryMCPS.txt	1.2k
Discoverier-new-agent.md	277
example_of_prompt_to_make_de-compresser_agent.txt	42
FirstFinalPlanToSpeedUp.md	774
GEMINI.md	1.2k
KIMI.md	1.4k
latest-folding-context-session.md	4
MessagingConnectorAssistant_Design.md	134
PIVOT_CHANGES.md	569
README.md	365
test-reloaded-session.md	4
TestsVisiblesAndVisibleExecutionFromClaude2Codex.md	230
Tlamatini-Kali-Setup.md	413
TLAMATINI_MCP.md	292
VERSIONING.md	470

docs/ · feature guides

32 files · 5,742 lines

agent-coverage.md	116	markdown-canvas.md	86
assets/readme-hero.svg	114	model-configuration-verification.md	44
build-root-assets-verification.md	162	model_configuration.md	338
canvas-and-interactor-quick-reference.txt	62	oob-namu-lifecycle.md	85
canvas-editing.md	183	pdf-canvas.md	260
canvas-refresh.md	110	prompting-flow-designer.md	121
chat-microphone-design.md	442	self-management-carriage.md	567
combined-release-builder.md	115	setup-lifecycle-research.md	129
companion-app-discovery.md	123	setup-wizards.md	257
dependency-coverage-audit.md	101	stm32er_all_families_proposal.md	171
desktop-input-and-flow-contracts.md	127	terminater.md	346
examples/prompting-kickoff.fpmt	26	torrenter.md	114
external_mcp_bulletproof_architecture.md	141	usage-dashboard.md	222
GUI-Manager-design.md	344	user-creation-wizard.md	99
interaction-settings.md	160	visual-analysis-errors.md	119
interactor.md	245	windows-flow-files.md	213

docs/claude/ · assistant onboarding (auto-imported)

10 files · 9,670 lines

acpx.md	458	exec-report.md	350
agents.md	409	frontend.md	576
architecture.md	941	gotchas.md	412
INDEX.md	156	recent-fixes.md	5.7k
mcp-tools.md	293		
multi-turn.md	394		

docs/changes/ · dated change records

29 files · 7,078 lines

2026-10-03-prompt-commentary-input.md	71
2026-10-03-prompt-commentary-input.patch	1.4k
2026-10-03-prompt-flow-documentation.md	59
2026-10-03-prompt-flow-documentation.patch	631
2026-10-04-commentary-graphical-redesign.md	55
2026-10-04-commentary-graphical-redesign.patch	2.0k
2026-10-04-prompt-flow-documentation.md	43
2026-10-04-release-validation.md	171
2026-10-04-run-output-resize.md	36
2026-10-04-run-output-resize.patch	834
2026-10-04-windows-flow-files.md	51
2026-10-07-latexer-verification.md	60
2026-10-07-pdf-mover-incident.md	87
2026-10-07-windows-access-verification.md	82
2026-10-10-composer-resize.md	80
2026-10-11-ui-sweep.md	146
2026-10-13-docked-avatar-divider.md	65
2026-10-16-v1.85.0.md	99
2026-10-18-v1.87.0.md	128
2026-10-19-private-release-data.md	97
2026-10-19-terminater.md	103
2026-10-20-terminater-audit.md	181
2026-10-20-v1.88.0.md	94
2026-10-22-documentation-refresh.md	76
2026-10-22-release-version-pinning.md	73
2026-10-22-v1.90.0.md	98
2026-10-23-v1.91.0.md	148
2026-10-27-v1.91.1.md	51
2026-11-01-v1.92.3.md	26

authoring guides

3 files · 2,057 lines

create a new agent, MCP/tool or skill

.agents/workflows/create_new_agent.md	753	.skills/create_new_skill.md	416
.mcps/create_new_mcp.md	888		

scripts/architecture_atlas/ and this atlas

4 files · 3,069 lines

generator and build script of Tlamatini-Architecture.pdf

scripts/architecture_atlas/build_architecture_pdf.py	238
scripts/architecture_atlas/gen_architecture.py	2.8k
scripts/architecture_atlas/README.md	54
Tlamatini-Architecture.pdf	2.1MB

.claude/ · Claude Code skills, hooks and settings

11 files · 3,728 lines

canonical skill set; SessionStart hook

hooks/announce_skills.py	204
hooks/test_announce_skills.py	353
settings.json	16
skills/tlamatini-agent-creation/SKILL.md	958
skills/tlamatini-agent-naming/SKILL.md	100
skills/tlamatini-daily-chat-test/.gitignore	8
skills/tlamatini-daily-chat-test/SKILL.md	278
skills/tlamatini-self-modify-inclusion/scripts/sweep_self_modify.py	436
skills/tlamatini-self-modify-inclusion/SKILL.md	354
skills/tlamatini-self-update-inclusion/scripts/sweep_self_update.py	580
skills/tlamatini-self-update-inclusion/SKILL.md	441

.claude/memory/ · persistent assistant memory

83 files · 3,235 lines

feedback, project and user notes

feedback_agent_naming_conventions.md	33
feedback_developer_product_boundary.md	18
feedback_dont_overbuild_exec_safety.md	29
feedback_forbidden_headless_visible_tests.md	20
feedback_foreground_visible_windows.md	19
feedback_hard_real_scenario_tests.md	25
feedback_main_branch_only.md	40
feedback_package_json_version_bump.md	32
feedback_run_tlamatini_agents_visible.md	34
feedback_state_constraints_upfront.md	31
feedback_track_changes_pivot_file.md	27
feedback_update_agent_docs.md	27
feedback_user_owns_git.md	43
MEMORY.md	151
project_acpx_oneshot_prompt.md	46
project_acpx_skills_menu.md	58
project_acpx_toggle_skills.md	35
project_acpxer_added.md	31
project_agent_table_wiped_on_boot.md	71
project_arduiner_agent.md	35
project_ask_execs_feature.md	35
project_audioplayer_agent.md	31
project_build_concurrency_guard.md	32
project_build_tests_and_installer_enterkey.md	25
project_camcorder_agent.md	27
project_carried_python_for_agents.md	37
project_conjunction_parser_fix.md	26
project_daily_chat_test_skill.md	28
project_desktop_ui_lifecycle.md	29
project_director_virtuoso_demo_prompts.md	28
project_doc_refresh_2026_05_06.md	40
project_doc_refresh_2026_05_08.md	39
project_doc_refresh_2026_05_09.md	58
project_eight_fixes_2026_05_07.md	53
project_emailer_recmailer_oneshot.md	36
project_embedding_memory_guard.md	35
project_esp32er_agent.md	37
project_execute_command_bounded_fix.md	27
project_execute_file_foreground_fix.md	44
project_external_exec_safety_layer.md	33
project_flow_compiler_dialog_wins.md	29
project_ghost_distinfo_dependency_audit.md	56
project_kalier_agent.md	37
project_keyboarder_wrapped_2026_05_07.md	32
project_latexer_agent.md	76
project_living_canvas_dropped.md	29
project_mouser_wrapped_2026_05_07.md	37
project_native_picker_tcltk_fix.md	57
project_native_toast.md	31
project_netspeed_calculator_agent.md	89
project_numpy_pyinstaller.md	24
project_ollama_source_build_breaks_embeddings.md	31
project_pdfer_agent.md	78
project_planner_followup.md	31
project_playwrighter_agent.md	31
project_playwrighter_hold_open.md	56
project_prompt_catalog_mode_badges.md	42
project_prompt_flow_commentary_input.md	19
project_pythonxer_forkbomb_fix.md	46
project_pythonxer_strict_ruff_gate.md	36
project_quote_apostrophe_fix_2026_05_07.md	48
project_reaper_console_window_fix.md	36
project_recorder_agent.md	31
project_reviewer_analyzer_agents.md	33
project_reviewer_analyzer_demo_prompts.md	25
project_reviewer_committed_secrets_falsepos.md	31
project_secret_leak_recovery.md	33
project_stm32er_agent.md	35
project_stm32er_bootstrap_preflight.md	31

.claude/memory/ · persistent assistant memory *(continued)*

83 files · 3,235 lines

feedback, project and user notes

project_stm32er_demo_prompts.md	29	project_temp_templates_policy.md	39
project_stm32er_docs_libs_parametrizer.md	35	project_unity_mcp_declined.md	30
project_stm32er_hil_serial_proof_fix.md	27	project_unrealer_expanded_surface.md	101
project_stm32er_tests.md	27	project_versioning_2026_05_15.md	71
project_summarizer_oneshot_mode.md	49	project_videoplayer_agent.md	33
project_taskbar_flash_attention.md	30	project_windower_agent.md	37
project_teletlamatini_acpx.md	34	user_profile.md	18

.gemini/skills/ · Gemini mirror

8 files · 3,155 lines

tlamatini-agent-creation/SKILL.md	958
tlamatini-agent-naming/SKILL.md	100
tlamatini-daily-chat-test/.gitignore	8
tlamatini-daily-chat-test/SKILL.md	278
tlamatini-self-modify-inclusion/scripts/sweep_self_modify.py	436
tlamatini-self-modify-inclusion/SKILL.md	354
tlamatini-self-update-inclusion/scripts/sweep_self_update.py	580
tlamatini-self-update-inclusion/SKILL.md	441

.codex/skills/ · Codex dossier and release skills

5 files · 546 lines

full-project-pdf-dossier/SKILL.md	150
overlap-safe-pptx-dossier/SKILL.md	161
tlamatini-release-validation/agents/openai.yaml	4
tlamatini-release-validation/references/acceptance-matrix.md	111
tlamatini-release-validation/SKILL.md	120

File index — 1638 tracked files

ID (click to see the tile) · path · [lines or size] · description extracted from the file itself. Grouped exactly like the atlas.

Atlas I — Server core: process, web tier and intelligence

Tlamatini/ · process entry — sheet 16

F1598 `Tlamatini/cat_art.py` [64 lines] *Determine the clear screen command based on the operating system*
F1603 `Tlamatini/manage.py` [1.7k lines] *Django's command-line utility for administrative tasks.*

Tlamatini/tlamatini/ · Django project — sheet 16

F1619 `Tlamatini/tlamatini/__init__.py` [10 lines] *Package marker for tlamatini/.*
F1620 `Tlamatini/tlamatini/asgi.py` [34 lines] *ASGI application: ProtocolTypeRouter sends HTTP to Django and WebSockets (AuthMiddlewareStack) to agent.routing.*
F1621 `Tlamatini/tlamatini/context_processors.py` [31 lines] *Defines static_version(), app_version().*
F1622 `Tlamatini/tlamatini/logging_filters.py` [31 lines] *Defines SuppressHttpGet200.*
F1623 `Tlamatini/tlamatini/middleware.py` [197 lines] *Wired to the console handler in settings.LOGGING ("tlamatini.request"), so everything logged here reaches BOTH the console and tlamatini.log (the…*
F1624 `Tlamatini/tlamatini/settings.py` [400 lines] *Django settings for tlamatini project.*
F1625 `Tlamatini/tlamatini/urls.py` [18 lines] *Module that sets urlpatterns using django, agent.views.*
F1626 `Tlamatini/tlamatini/wsgi.py` [28 lines] *Standard Django WSGI entry point (the app itself runs on ASGI/Daphne).*

agent/ · transport and web endpoints — sheet 16

F0474 `Tlamatini/agent/__init__.py` [26 lines] *Ensure absolute imports for generated protobuf modules resolve when code tries to import them as top-level names.*
F0475 `Tlamatini/agent/access_key_wizard.py` [703 lines] *Backend helpers for the graphical Access Keys Wizard.*
F0487 `Tlamatini/agent/admin.py` [31 lines] *Module importing django, models.*
F0747 `Tlamatini/agent/apps.py` [838 lines] *Companion-app discovery (Tlamatini-FlowPills) — process-local gate Dedicated to discovery publication and DELIBERATELY separate from…*
F0756 `Tlamatini/agent/chat_voice_consumer.py` [121 lines] *Authenticated direct dictation, isolated from model/chat orchestration.*
F0764 `Tlamatini/agent/constants.py` [156 lines] *Tlamatini speaks in the FIRST PERSON (Angela, 2026-10-09) Every fixed line below is a chat message FROM Tlamatini, and the chat…*
F0765 `Tlamatini/agent/consumers.py` [2.5k lines] *Defines AgentConsumer.*
F0766 `Tlamatini/agent/contacts.py` [210 lines] *Tlamatini Contacts book — resolve a person's NAME to their messaging handles.*
F0790 `Tlamatini/agent/flow_file_views.py` [50 lines] *Authenticated browser handoff and loopback instance identification.*
F0830 `Tlamatini/agent/model_brain_views.py` [133 lines] *HTTP surface of the MODEL BRAIN's Auto-tuning dialog (Config -> Models -> Save).*
F0834 `Tlamatini/agent/native_dialogs.py` [504 lines] *Tkinter-free native file/folder pickers for the Tlamatini server.*
F0843 `Tlamatini/agent/pdf_context_views.py` [40 lines] *Authenticated multipart preparation of a PDF selected in the browser canvas.*
F0846 `Tlamatini/agent/prompt_flow_panel_consumer.py` [93 lines] *Authenticated, connection-scoped playback; never infer completion from prose.*
F0864 `Tlamatini/agent/routing.py` [24 lines] *Module that sets websocket_urlpatterns using django, prompt_flow_panel_consumer, chat_voice_consumer, terminal_consumer.*
F1587 `Tlamatini/agent/urls.py` [230 lines] *Defines secure_get(), secure_post().*
F1590 `Tlamatini/agent/usage_views.py` [55 lines] *Authenticated, read-only usage; independent of the chat operation lock.*
F1591 `Tlamatini/agent/version.py` [388 lines] *Version resolution for Tlamatini.*
F1592 `Tlamatini/agent/views.py` [13k lines] *Defines home(), login_view(), welcome_view(), logout_view() and 173 more.*
F1596 `Tlamatini/agent/window_flash.py` [146 lines] *Flash the Tlamatini.exe console window's taskbar button + log an UPPERCASE attention banner when a browser surface needs the user.*

agent/templates/ · server-rendered pages — sheet 16

F1380 `Tlamatini/agent/templates/admin/base_site.html` [15 lines] *HTML template. Loads 1 scripts and 3 stylesheets.*
F1381 `Tlamatini/agent/templates/agent/agent_page.html` [949 lines] *dialog_theme.css MUST stay LAST: it supersedes the legacy .ui-dialog chrome on cascade ORDER. Loads 66 scripts and 27 stylesheets.*
F1382 `Tlamatini/agent/templates/agent/agentic_control_panel.html` [358 lines] *dialog_theme.css MUST stay LAST: it supersedes the legacy .ui-dialog chrome on cascade ORDER. Loads 33 scripts and 12 stylesheets.*
F1383 `Tlamatini/agent/templates/agent/interaction_settings.html` [42 lines] *HTML template.*
F1384 `Tlamatini/agent/templates/agent/interactor.html` [23 lines] *HTML template.*
F1385 `Tlamatini/agent/templates/agent/login.html` [63 lines] *Page "Login - Tlamatini". Loads 2 scripts and 8 stylesheets.*
F1386 `Tlamatini/agent/templates/agent/prompt_flow_panel.html` [131 lines] *Shared policy first: Escape = Cancel; clicking outside keeps dialogs open. Loads 16 scripts and 13 stylesheets.*
F1387 `Tlamatini/agent/templates/agent/torrenter_sources.html` [25 lines] *Page "Torrenter Sources · Tlamatini".*
F1388 `Tlamatini/agent/templates/agent/usage_dialog.html` [64 lines] *HTML template.*
F1389 `Tlamatini/agent/templates/agent/welcome.html` [42 lines] *Page "Welcome - Tlamatini". Loads 4 scripts and 9 stylesheets.*

agent/management/ · manage.py commands — sheet 16

F0816 `Tlamatini/agent/management/__init__.py` [10 lines] *Django management package for the agent app.*
F0817 `Tlamatini/agent/management/commands/acpx_lint.py` [123 lines] *Django management command to lint every SKILL.md package in agent/skills_pkg/.*
F0818 `Tlamatini/agent/management/commands/check_agent_runtimes.py` [109 lines] *Offline runtime-carriage check, executed in source and by the frozen build.*
F0819 `Tlamatini/agent/management/commands/check_prompt_flow_panel.py` [150 lines] *Offline panel checks for source and real frozen releases.*
F0820 `Tlamatini/agent/management/commands/check_terminator.py` [227 lines] *Offline Terminals / Terminator checks for source and real frozen releases.*
F0821 `Tlamatini/agent/management/commands/enroll_torrenter.py` [29 lines] *Defines Command.*
F0822 `Tlamatini/agent/management/commands/startserver.py` [56 lines] *Defines Command.*

agent/ · intelligence core — sheet 16

F0489 `Tlamatini/agent/agent_verdict.py` [657 lines] *Deterministic verdict engine for wrapped-agent results.*
F0748 `Tlamatini/agent/cancellation.py` [199 lines] *agent/cancellation.py — the PER-RUN cancellation latch.*
F0750 `Tlamatini/agent/capability_registry.py` [691 lines] *Defines ToolCapability, ContextCapability, build_context_capabilities(), build_tool_capabilities() and 2 more.*
F0753 `Tlamatini/agent/chat_agent_registry.py` [2.4k lines] *Defines ChatWrappedAgentSpec, agent_row_name(), get_wrapped_agent_security_hints().*
F0754 `Tlamatini/agent/chat_agent_runtime.py` [638 lines] *Defines get_chat_runtime_root(), ensure_chat_runtime_root(), create_isolated_runtime_copy(), resolve_runtime_script_path() and 11 more.*
F0755 `Tlamatini/agent/chat_history_loader.py` [117 lines] *DB-backed chat history loader (safe, transparent)*
F0757 `Tlamatini/agent/chat_voice_runtime.py` [242 lines] *Direct resident Whisperer bridge.*
F0758 `Tlamatini/agent/chat_voice_settings.py` [70 lines] *Small, dependency-free contract shared by Django and carried dictation Python.*
F0761 `Tlamatini/agent/compact_mode.py` [685 lines] *compact_mode.py - the COMPACT MODE switch (Angela, 2026-10-02).*
F0762 `Tlamatini/agent/config.json` [299 lines] *Main configuration template: models, ports, timeouts, Model Brain, Compact mode, binary guard, ACPX; secrets are placeholders.*
F0763 `Tlamatini/agent/config_loader.py` [135 lines] *Defines find_config_path(), load_config(), get_config_value(), get_int_config_value() and 1 more.*
F0767 `Tlamatini/agent/context_baseline.py` [489 lines] *Tlamatini - the context gauge AT REST.*
F0768 `Tlamatini/agent/context_fitter.py` [760 lines] *Tlamatini - the CONTEXT FITTER: every request fits the model it is sent to.*
F0769 `Tlamatini/agent/context_governor.py` [2.0k lines] *Tlamatini - the CONTEXT GOVERNOR.*
F0781 `Tlamatini/agent/embedding_memory_guard.py` [433 lines] *Embedding-memory pre-flight guard for "Set directory as context".*
F0782 `Tlamatini/agent/exec_permission.py` [259 lines] *Ask-Execs permission broker.*
F0791 `Tlamatini/agent/global_execution_planner.py` [526 lines] *Defines GlobalPlanNode, GlobalExecutionPlan, build_global_execution_plan(), summarize_global_execution_plan() and 2 more.*
F0792 `Tlamatini/agent/global_state.py` [53 lines] *Defines GlobalState, get_request_state(), scoped_request_state().*
F0793 `Tlamatini/agent/gpu_perf.py` [420 lines] *GPU max-performance + Ollama-pinning startup hook for Tlamatini.*
F0802 `Tlamatini/agent/inet_determiner.py` [219 lines] *Defines determine_internet_required().*
F0813 `Tlamatini/agent/llm_timing.py` [160 lines] *Ollama call timing instrumentation.*
F0814 `Tlamatini/agent/local_toolcall_parser.py` [376 lines] *Recover a tool call that a LOCAL model emitted as PLAIN TEXT.*
F0823 `Tlamatini/agent/mcp_agent.py` [4.1k lines] *MCP Agent (mcp_agent.py)*
F0828 `Tlamatini/agent/model_brain.py` [549 lines] *MODEL BRAIN - Tlamatini configures herself for EACH model, from formal sources.*
F0829 `Tlamatini/agent/model_brain_chat.py` [178 lines] *The chat model the Model Brain configures (agent/model_brain.py).*
F0831 `Tlamatini/agent/model_profiles.json` [102 lines] *Model Brain knowledge base: sourced sampling/thinking profiles per model family.*
F0835 `Tlamatini/agent/ollama_stop.py` [68 lines] *Stop sequences every Ollama model will accept.*
F0841 `Tlamatini/agent/pdf_context.py` [202 lines] *PDF canvas ingestion: complete text and optional images for the normal RAG chain.*
F0842 `Tlamatini/agent/pdf_context_jobs.py` [113 lines] *Bounded background jobs for PDF extraction and Image-Interpreter analysis.*
F0844 `Tlamatini/agent/pdf_image_analysis.py` [38 lines] *PDF-specific adapter for the existing Image-Interpreter agent's vision pipeline.*
F0845 `Tlamatini/agent/prompt.pmt` [401 lines] *System prompt of the chat LLM: Tlamatini's identity, operating rules, tool-usage contract, Temp/Templates policy and the self-knowledge slot.*
F0847 `Tlamatini/agent/prompt_flow_panel_runtime.py` [207 lines] *Model adapter for a single prompt-flow-panel run, using the existing RAG stack.*
F0863 `Tlamatini/agent/rag_enhancements.py` [687 lines] *RAG Enhancement Module for Tlamatini Provides advanced metadata extraction, context optimization, and retrieval strategies.*
F0867 `Tlamatini/agent/self_healing.py` [523 lines] *agent/self_healing.py — Self-aware, NEVER-HANGING, self-healing resilience for Tlamatini's Multi-Turn model steps.*
F1577 `Tlamatini/agent/Tlamatini.md` [550 lines] *Tlamatini — Self-Knowledge: The combined builder predicts both editions' actual emitted version before building, pins one matching number, runs Private/Public…*
F1579 `Tlamatini/agent/tools.py` [5.3k lines] *Defines launch_in_new_terminal(), _ConfigRequirementAnalyzer, get_all_agents(), get_all_agent_processes() and 25 more.*
F1588 `Tlamatini/agent/usage_provider.py` [333 lines] *Bounded, read-only Ollama usage adapter.*
F1589 `Tlamatini/agent/usage_tracking.py` [108 lines] *Metadata-only usage ledger; no prompts, responses or credentials are stored.*
F1593 `Tlamatini/agent/visual_error_reporting.py` [29 lines] *Per-request visual-error reporting, independent of execution/recovery policy.*
F1594 `Tlamatini/agent/web_search_llm.py` [279 lines] *Updated web_search_llm.py to allow custom Ollama server IP and port*

agent/interactor/ - live editor module — sheet 16

F0805 `Tlamatini/agent/interactor/__init__.py` [2 lines] *Interactor: the live canvas bridge and guided editing capability (not an agent).*
F0806 `Tlamatini/agent/interactor/consumer.py` [74 lines] *Session-authenticated, same-origin editor transport.*
F0807 `Tlamatini/agent/interactor/context.py` [35 lines] *The originating tab is request-scoped, never a global 'active editor'.*
F0808 `Tlamatini/agent/interactor/editor.mjs` [154 lines] *CodeMirror extension: field guidance, optional ghost text and observable typing.*
F0809 `Tlamatini/agent/interactor/files.py` [49 lines] *Host files are opened only by the model's scoped tool, never a browser path frame.*
F0810 `Tlamatini/agent/interactor/hub.py` [128 lines] *Bounded request/receipt bridge between synchronous tools and an authenticated tab.*
F0811 `Tlamatini/agent/interactor/protocol.py` [56 lines] *Small, explicit wire contract.*
F0812 `Tlamatini/agent/interactor/tool.py` [146 lines] *One typed model tool, no workflow Agent or worker process.*

agent/rag/ · retrieval-augmented generation — sheet 16

F0848 `Tlamatini/agent/rag/__init__.py` [39 lines] *Module that sets __all__ using factory, interface, chains.*
F0849 `Tlamatini/agent/rag/binary_guard.py` [516 lines] *Binary-content detection for the RAG context-loading pipeline.*
F0854 `Tlamatini/agent/rag/config.py` [445 lines] *The LLM's self-knowledge file.*
F0855 `Tlamatini/agent/rag/factory.py` [1.2k lines] *Defines get_system_context_sync(), get_files_context_sync(), build_prompt_only_chain(), build_retrieval_chain() and 3 more.*
F0856 `Tlamatini/agent/rag/interaction.py` [67 lines] *Defines save_context_blob(), load_context_blob(), show_rephrased_question().*
F0857 `Tlamatini/agent/rag/interface.py` [967 lines] *Defines get_program_by_name(), tokenCounterOfAsk(), is_valid_prompt(), ask_rag().*
F0858 `Tlamatini/agent/rag/loaders.py` [51 lines] *Defines report_oversized_docs().*
F0859 `Tlamatini/agent/rag/prompts.py` [33 lines] *Contextualization prompt for history-aware question rewriting*
F0860 `Tlamatini/agent/rag/retrieval.py` [114 lines] *Defines retrieve_documents().*
F0861 `Tlamatini/agent/rag/splitters.py` [13 lines] *Defines get_text_splitter().*
F0862 `Tlamatini/agent/rag/utils.py` [191 lines] *Scope header prepended to the user-loaded context blob before it is bound to the prompt's {context} placeholder (history-aware RAG chain) or folded…*

agent/rag/chains/ · answer chains — sheet 16

F0850 `Tlamatini/agent/rag/chains/base.py` [177 lines] *Defines GenerationCancelledException, Callbacks, summarize_or_none().*
F0851 `Tlamatini/agent/rag/chains/basic.py` [193 lines] *Defines BasicPromptOnlyChain.*
F0852 `Tlamatini/agent/rag/chains/history_aware.py` [587 lines] *Regex to strip code blocks (or ") so embedded code doesn't cause false-positive file-listing detection.*
F0853 `Tlamatini/agent/rag/chains/unified.py` [1.1k lines] *Defines wrap_loaded_context(), wrap_retrieved_context(), with_system_context(), history_summary_tail() and 3 more.*

agent/services/ · flow compiler and contracts — sheet 16

F0869 `Tlamatini/agent/services/__init__.py` [25 lines] *Pure format validators are also used before Django starts (Explorer opening).*
F0870 `Tlamatini/agent/services/agent_contracts.py` [501 lines] *Defines AgentContract, get_agent_contracts(), get_password_paths(), resolve_agent_type() and 5 more.*
F0871 `Tlamatini/agent/services/agent_paths.py` [178 lines] *Defines is_frozen_mode(), get_agent_app_dir(), get_app_base_dir(), get_agents_root() and 9 more.*
F0872 `Tlamatini/agent/services/filesystem.py` [141 lines] *Defines get_time_stamp(), generate_tree_view_content(), save_files_from_db().*
F0873 `Tlamatini/agent/services/flow_compiler.py` [441 lines] *Defines FlowCompileError, _QuotedStr, _AgentConfigYamlDumper, dump_agent_config_yaml() and 3 more.*
F0874 `Tlamatini/agent/services/flow_knowledge.py` [147 lines] *Export the installed agent contracts for standalone planning/monitoring agents.*
F0875 `Tlamatini/agent/services/flow_spec.py` [174 lines] *Defines FlowNode, FlowConnection, FlowSpec, normalize_flow_payload() and 1 more.*
F0876 `Tlamatini/agent/services/prompt_flow_panel.py` [438 lines] *Versioned .fpmt graph contract and deterministic prompt-flow-panel interpreter.*
F0877 `Tlamatini/agent/services/response_parser.py` [678 lines] *Full-answer debug dumps (the ENTIRE LLM response used to be printed up to 3x per turn below) are expensive and noisy on long answers.*

agent/imaging/ · image analysis — sheet 16

F0799 `Tlamatini/agent/imaging/converter.py` [25 lines] *Defines convert_image_to_base64().*
F0800 `Tlamatini/agent/imaging/image_interpreter.py` [399 lines] *Defines opus_analyze_image(), qwen_analyze_image().*
F0801 `Tlamatini/agent/imaging/screenshot.py` [257 lines] *screenshot.py - Capture the full screen and save as a .jpg file.*

agent/opus_client/ · Claude API client — sheet 16

F0836 `Tlamatini/agent/opus_client/claude_opus_client.py` [969 lines] *Claude Opus 4.5 Python Client A comprehensive client for interacting with Claude Opus 4.5 via the Anthropic API.*
F0837 `Tlamatini/agent/opus_client/examples.py` [9 lines] *Python module.*
F0838 `Tlamatini/agent/opus_client/README.md` [418 lines] *Claude Opus 4.5 Python Client: A comprehensive, production-ready Python client for interacting with Claude Opus 4.5 via the Anthropic API.*

agent/monitors/ · SQL monitor — sheet 16

F0833 `Tlamatini/agent/monitors/monitor_sql.py` [83 lines] *CONFIGURATION Windows typically uses the "ODBC Driver 17 for SQL Server" Ensure you have installed the driver from Microsoft.*

agent/doc_generation/ · project dossier generator — sheet 16

F0771 `Tlamatini/agent/doc_generation/complete_project_docs.py` [639 lines] *Complete project dossier: facts, orchestration and verification.*
F0772 `Tlamatini/agent/doc_generation/dossier_content.py` [2.6k lines] *The written content of the project dossier, shared by the PDF and the deck.*
F0773 `Tlamatini/agent/doc_generation/dossier_editor_content.py` [295 lines] *Canvas, Interactor and configuration chapters shared by the PDF and deck.*
F0774 `Tlamatini/agent/doc_generation/dossier_pdf.py` [1.5k lines] *The printed dossier: an A4 codex with a dark cover, dark chapter openers and ivory reading pages.*
F0775 `Tlamatini/agent/doc_generation/dossier_pptx.py` [1.2k lines] *The dossier deck: a 16:9 obsidian-and-jade presentation built from the same content as the PDF.*
F0776 `Tlamatini/agent/doc_generation/dossier_theme.py` [385 lines] *Shared visual identity of the project dossier (PDF + PPTX).*
F0777 `Tlamatini/agent/doc_generation/dossier_verify.py` [314 lines] *Re-open the finished dossier files and prove they are whole.*
F0778 `Tlamatini/agent/doc_generation/mardown_to_pdf.py` [119 lines] *Defines markdown_text_to_pdf(), main().*
F0779 `Tlamatini/agent/doc_generation/refresh_project_docs.py` [16 lines] *Module importing __future__, complete_project_docs.*
F0780 `Tlamatini/agent/doc_generation/test_output.pdf` [41KB] *PDF document: test output.*

agent/ · other modules — sheet 16

F0749 `Tlamatini/agent/canvas_file_views.py` [331 lines] *Native canvas saves: a selected destination is an expiring capability.*
F0803 `Tlamatini/agent/interaction_narration.py` [131 lines] *Bounded narration receipts.*
F0804 `Tlamatini/agent/interaction_settings.py` [162 lines] *Account preferences for real editor/terminal actions; presentation stays local.*
F1390 `Tlamatini/agent/terminal_consumer.py` [271 lines] *ws/terminals/ – the browser side of Terminator.*
F1391 `Tlamatini/agent/terminal_guard.py` [483 lines] *terminal_guard.py – honesty rails for live terminal tutorials (Terminator).*
F1392 `Tlamatini/agent/terminal_hub.py` [347 lines] *The Django-side home of the Terminator terminals.*
F1393 `Tlamatini/agent/terminal_views.py` [69 lines] *HTTP surface of Terminator.*
F1394 `Tlamatini/agent/terminator_testkit.py` [81 lines] *Shared helpers for the Terminator test suites (not a test module itself).*
F1395 `Tlamatini/agent/terminator_tool.py` [429 lines] *terminator_tool.py – the TYPED door from the chat model to Terminator.*
F1580 `Tlamatini/agent/torrenter_bridge.py` [116 lines] *Only authenticated server launch sites can bind a Jobs profile to a pool.*
F1581 `Tlamatini/agent/torrenter_tool.py` [64 lines] *The model gets actions and explicit arguments.*
F1582 `Tlamatini/agent/torrenter_views.py` [220 lines] *Stage private torrent metadata directly from the authenticated browser to Jobs.*

Atlas II — Integration surfaces, platform and state

agent/ · context providers and external-MCP client — sheet 17

F0751 `Tlamatini/agent/chain_files_search_lcel.py` [791 lines] *Handle both relative import from the same directory (when used as module) and direct import (when run as script)*
F0752 `Tlamatini/agent/chain_system_lcel.py` [282 lines] *Handle both relative import (when used as module) and direct import (when run as script)*
F0759 `Tlamatini/agent/command_togen_pb2.txt` [32 lines] *Command used to regenerate the Files-Search protobuf/gRPC stubs.*
F0770 `Tlamatini/agent/context_sidecar_timeout.py` [163 lines] *A time limit for the context sidecars' own Ollama calls.*
F0783 `Tlamatini/agent/external_mcp_defaults.py` [245 lines] *Default External MCP servers — shipped with EVERY installation, INACTIVE.*
F0784 `Tlamatini/agent/external_mcp_manager.py` [2.7k lines] *External MCP client layer.*
F0785 `Tlamatini/agent/external_mcps.json` [4 lines] *External-MCP catalog (mcpServers shape) + active list; user state preserved across self-update.*
F0786 `Tlamatini/agent/filesearch.proto` [33 lines] *gRPC contract of the Files-Search MCP service.*
F0787 `Tlamatini/agent/filesearch_pb2.py` [57 lines] *Generated protocol buffer code.*
F0788 `Tlamatini/agent/filesearch_pb2_grpc.py` [152 lines] *Client and server classes corresponding to protobuf-defined services.*
F0824 `Tlamatini/agent/mcp_files_search_client.py` [462 lines] *Config loading utilities*
F0825 `Tlamatini/agent/mcp_files_search_server.py` [240 lines] *Defines get_application_root(), load_config(), FileSearcherServicer, serve().*
F0826 `Tlamatini/agent/mcp_system_client.py` [141 lines] *Defines MCPSystemClient, example().*
F0827 `Tlamatini/agent/mcp_system_server.py` [235 lines] *Defines MCPSystemHandler, system_handler(), main().*
F0866 `Tlamatini/agent/runtime_provisioner.py` [1.2k lines] *Runtime Provisioner — Tlamatini ALWAYS has node / npm / npx / pnpm / uv / uvx.*
F1595 `Tlamatini/agent/win_shim.py` [189 lines] *Windows command resolution — the ONE definition of "how do we spawn this?".*

agent/acpx/ · ACPX runtime — sheet 17

F0476 `Tlamatini/agent/acpx/__init__.py` [135 lines] *ACPX runtime for Tlamatini.*
F0477 `Tlamatini/agent/acpx/agent_registry.py` [393 lines] *ACP agent registry — the agent_id -> command map.*
F0478 `Tlamatini/agent/acpx/child_health.py` [299 lines] *ACPX child-health classification — did the external CLI actually DELIVER?*
F0479 `Tlamatini/agent/acpx/config.py` [549 lines] *ACPX plugin config — Python mirror of OpenClaw's extensions/acpx/src/config-schema.ts and config.ts.*
F0480 `Tlamatini/agent/acpx/permissions.py` [108 lines] *Permission gate for ACPX runtime.*
F0481 `Tlamatini/agent/acpx/runtime.py` [1.3k lines] *ACPX runtime — Python port of OpenClaw's BaseAcpxRuntime + AcpRuntime (extensions/acpx/src/runtime.ts).*
F0482 `Tlamatini/agent/acpx/service.py` [112 lines] *Django-side ACPX service — registers the runtime at startup and mirrors disk state into the AcpAgent table.*
F0483 `Tlamatini/agent/acpx/session_store.py` [145 lines] *ACP session store — Python port of OpenClaw's createFileSessionStore() + createResetAwareSessionStore() in extensions/acpx/src/runtime.ts.*
F0484 `Tlamatini/agent/acpx/tests.py` [1.6k lines] *Sanity tests for the ACPX runtime + Skill harness. [98 tests]*
F0485 `Tlamatini/agent/acpx/tools.py` [716 lines] *ACPX-related LangChain @tool functions exposed to the unified agent.*
F0486 `Tlamatini/agent/acpx/windows_spawn.py` [180 lines] *Windows spawn helpers — Python port of OpenClaw's applyWindowsSpawnProgramPolicy() / materializeWindowsSpawnProgram() /…*

agent/skills/ · skill harness — sheet 17

F0878 `Tlamatini/agent/skills/__init__.py` [29 lines] *Tlamatini skills package — runtime side.*
F0879 `Tlamatini/agent/skills/frontmatter.py` [186 lines] *SKILL.md frontmatter parser.*
F0880 `Tlamatini/agent/skills/harness.py` [603 lines] *SkillHarness — one skill invocation: contract validation, secret redaction, audit logging, and delivery of the skill's procedure to whoever will run…*
F0881 `Tlamatini/agent/skills/io_contract.py` [147 lines] *Input / output contract validation for skills.*
F0882 `Tlamatini/agent/skills/redaction.py` [234 lines] *Secret redaction for the skill boundary — the ONE definition of "is this input a credential, and what may leave the harness".*
F0883 `Tlamatini/agent/skills/registry.py` [295 lines] *Skill registry — discovers, parses, and caches all SKILL.md packages.*
F0884 `Tlamatini/agent/skills/validation.py` [335 lines] *The ONE validator for a SKILL.md package.*

agent/skills_pkg/ · SKILL.md packages — sheet 17

F0885 `Tlamatini/agent/skills_pkg/_meta/lint.py` [85 lines] *Lint every SKILL.md in agent/skills_pkg/.*
F0886 `Tlamatini/agent/skills_pkg/_meta/schema.json` [101 lines] *JSON document with keys: $schema, title, description, type, required, additionalProperties, properties.*
F0887 `Tlamatini/agent/skills_pkg/acp_router/SKILL.md` [106 lines] *Route plain-language requests for Claude Code, Cursor, Codex, Gemini, Qwen, Pi, Kiro, Kimi, iFlow, Factory Droid, Kilocode, OpenCode, Copilot into…*
F0888 `Tlamatini/agent/skills_pkg/adding_external_mcp/references/external_mcp_catalog_format.md` [127 lines] *External MCP Catalog Format — Detailed Guide: The external MCP catalog lives in external_mcps.json next to config.json.*
F0889 `Tlamatini/agent/skills_pkg/adding_external_mcp/references/llm_reflection_research.md` [112 lines] *LLM Reflection — Research Findings: LLM Reflection is a technique where language models evaluate and improve their own outputs through iterative…*
F0890 `Tlamatini/agent/skills_pkg/adding_external_mcp/references/transport_guide.md` [143 lines] *Transport Configuration Reference: Tlamatini supports four MCP transport types via _SUPPORTED_TRANSPORTS*
F0891 `Tlamatini/agent/skills_pkg/adding_external_mcp/references/troubleshooting.md` [164 lines] *Troubleshooting and Diagnostics: Run a static triage BEFORE activating a server*
F0892 `Tlamatini/agent/skills_pkg/adding_external_mcp/SKILL.md` [297 lines] *> The authoritative reference for adding a NEW external MCP server to Tlamatini universal MCP client layer.*
F0893 `Tlamatini/agent/skills_pkg/code_review/SKILL.md` [133 lines] *Review a git diff (or working-tree changes) like a senior engineer — correctness, security, performance, readability — and return a verdict plus…*
F0894 `Tlamatini/agent/skills_pkg/create_new_agent/SKILL.md` [156 lines] *The authoritative 8-step contract for scaffolding a NEW Tlamatini workflow agent end-to-end (backend script + config.yaml, connection-update view +…*
F0895 `Tlamatini/agent/skills_pkg/create_new_mcp/SKILL.md` [147 lines] *The authoritative reference for adding a NEW tool, MCP context provider, or BOTH to Tlamatini.*
F0896 `Tlamatini/agent/skills_pkg/flow_making/references/flw_schema.md` [89 lines] *.flw schema (schemaVersion 2) — the shape the canvas loader consumes: Source of truth: agent/static/agent/js/acp-file-io.js::… and…*
F0897 `Tlamatini/agent/skills_pkg/flow_making/scripts/make_flow.py` [178 lines] *make_flow.py — one-shot driver for the flow-making skill.*
F0898 `Tlamatini/agent/skills_pkg/flow_making/scripts/result_to_flw.py` [236 lines] *result_to_flw.py — convert FlowCreator's flow_result.json into a canonical .flw flow file that the Tlamatini ACP canvas loader (loadDiagram in…*
F0899 `Tlamatini/agent/skills_pkg/flow_making/SKILL.md` [164 lines] *Turn a natural-language objective into a downloadable .flw workflow by driving the FlowCreator engine (full 86-agent catalog), then emit a…*
F0900 `Tlamatini/agent/skills_pkg/github/SKILL.md` [78 lines] *Use the gh CLI for GitHub issues, PR status, CI/logs, comments, reviews, releases, and API queries.*
F0901 `Tlamatini/agent/skills_pkg/gmail/SKILL.md` [59 lines] *Read and send Gmail messages and threads via Gmail API.*
F0902 `Tlamatini/agent/skills_pkg/hello_world/SKILL.md` [54 lines] *Smoke-test skill that echoes its inputs and confirms the SkillHarness wiring is alive end-to-end.*
F0903 `Tlamatini/agent/skills_pkg/jira/SKILL.md` [65 lines] *Read and write Jira issues, comments, transitions via the Jira REST API v3.*
F0904 `Tlamatini/agent/skills_pkg/kali_pentest/SKILL.md` [128 lines] *Drive an authorized, scoped penetration-testing / recon / CTF workflow against a target through the Kalier agent and the MCP-Kali-Server —…*
F0905 `Tlamatini/agent/skills_pkg/notion/SKILL.md` [77 lines] *Use the Notion API to create, read, update Notion pages, data sources (databases), and blocks.*
F0906 `Tlamatini/agent/skills_pkg/roblox_studio/SKILL.md` [177 lines] *Build and edit in Roblox Studio via the Roblox Studio MCP the RIGHT way - preflight the Studio connection, do the whole build in a few big…*
F0907 `Tlamatini/agent/skills_pkg/security_audit/SKILL.md` [113 lines] *Run whichever SAST, secret-scanning, and dependency-audit tools are installed (bandit, semgrep, ruff, eslint, gitleaks, pip-audit) over a path, then…*

F0908 `Tlamatini/agent/skills_pkg/setup_new_acpx_key/SKILL.md` [274 lines] *Configure the API key (or other credential) for a registered ACPX agent_id. . .*
F0909 `Tlamatini/agent/skills_pkg/skill_creator/scripts/quick_validate.py` [63 lines] *Validate a single SKILL.md package directory.*
F0910 `Tlamatini/agent/skills_pkg/skill_creator/SKILL.md` [129 lines] *Create, edit, validate, and package new Tlamatini skills (SKILL.md packages).*
F0911 `Tlamatini/agent/skills_pkg/slack/SKILL.md` [60 lines] *Post messages, read channels, and manage threads via Slack Web API.*
F0912 `Tlamatini/agent/skills_pkg/summarize/SKILL.md` [60 lines] *Summarize a long text or file into a tight, faithful brief at a target word count.*
F0913 `Tlamatini/agent/skills_pkg/tlamatini_allowed_hosts_tighten/SKILL.md` [64 lines] *Tighten Django ALLOWED_HOSTS from the wide-open '*' default to a specific list, with a backup of settings.py.*
F0914 `Tlamatini/agent/skills_pkg/tlamatini_csrf_exempt_audit/SKILL.md` [100 lines] *Enumerate every @csrf_exempt-decorated view in Tlamatini/agent/views.py and classify whether each one really needs the exemption.*
F0915 `Tlamatini/agent/skills_pkg/tlamatini_exec_report_row_adder/SKILL.md` [85 lines] *Add a state-changing tool to _EXEC_REPORT_TOOLS in mcp_agent.py and the matching CSS rules so its operations appear in the chat-page Exec Report.*
F0916 `Tlamatini/agent/skills_pkg/tlamatini_flow_from_objective/SKILL.md` [126 lines] *Turn a one-sentence objective into a downloadable .flw workflow that wires the right Tlamatini visual agents and connections.*
F0917 `Tlamatini/agent/skills_pkg/tlamatini_flw_doctor/SKILL.md` [130 lines] *Validate a .flw workflow file: check connection topology, terminal agents, Parametrizer single-lane queue, missing connectors, dangling target_agents.*
F0918 `Tlamatini/agent/skills_pkg/tlamatini_new_acp_agent/SKILL.md` [146 lines] *Scaffold a new Tlamatini visual agent end-to-end across the 8 places the agent contract requires (script + config.yaml, view + url, migration, CSS. . .*
F0919 `Tlamatini/agent/skills_pkg/tlamatini_planner_trace_replay/SKILL.md` [85 lines] *Replay the most recent global execution planner trace from tlamatini.log and explain why each capability got the score it did.*
F0920 `Tlamatini/agent/skills_pkg/tlamatini_static_version_bumper/SKILL.md` [103 lines] *Bump STATIC_VERSION in tlamatini/settings.py so the chat / ACP frontend re-fetches CSS+JS bundles after a frontend change.*
F0921 `Tlamatini/agent/skills_pkg/todoist/SKILL.md` [58 lines] *Manage Todoist tasks and projects via the Todoist REST API v2.*
F0922 `Tlamatini/agent/skills_pkg/trello/SKILL.md` [57 lines] *Manage Trello boards, lists, and cards via the Trello REST API.*
F0923 `Tlamatini/agent/skills_pkg/weather/SKILL.md` [57 lines] *Look up current weather and short forecast for a location via Open-Meteo (no API key required).*

agent/ · platform, safety and lifecycle sheet 17

F0488 `Tlamatini/agent/agent_manifest.py` [277 lines] *Companion-app discovery: the agents manifest + the HKCU discovery key.*
F0760 `Tlamatini/agent/command_watchdog.py` [602 lines] *Autonomous command watchdog for Tlamatini.*
F0789 `Tlamatini/agent/flow_file_open.py` [312 lines] *Bounded .flw / .fpmt opening across Explorer, startup, login and browser uploads.*
F0815 `Tlamatini/agent/log_identity.py` [358 lines] *Per-line USER attribution for tlamatini.log (Angela, 2026-08-13).*
F0839 `Tlamatini/agent/orphan_reaper.py` [895 lines] *Orphan-process reaper for Tlamatini.*
F0840 `Tlamatini/agent/path_guard.py` [407 lines] *path_guard.py — Centralized path validation for LLM-exposed tools.*
F0868 `Tlamatini/agent/self_update.py` [500 lines] *Tlamatini self-update — "About ▷ Check for updates".*
F1597 `Tlamatini/agent/windows_app_registration.py` [285 lines] *Register Tlamatini in Windows "Installed apps" / "Programs and Features".*

persistence core sheet 17

F0832 `Tlamatini/agent/models.py` [310 lines] *Defines AgentMessage, LLMProgram, LLMSnippet, Prompt and 17 more.*
F0865 `Tlamatini/agent/run_db_wal_tests.ps1` [54 lines] *Tlamatini — VISIBLE test runner for the DB Backup / Set DB (WAL) fix.*
F0924 `Tlamatini/agent/sqlite_copy.py` [407 lines] *WAL-safe SQLite copying + sidecar hygiene — what "Backup database" and "Set DB" must use instead of a plain file copy.*
F1599 `Tlamatini/DB/Older/README.md` [60 lines] *DB / Older: This directory is the archive of previously-live db.sqlite3 files that the start-up swap-in has retired.*
F1600 `Tlamatini/DB/ToLoad/README.md` [63 lines] *DB / ToLoad: This directory is the drop zone for replacing Tlamatini's live SQLite database on the next start-up.*

Tlamatini/jd-cli/ · bundled Java decompiler sheet 17

F1601 `Tlamatini/jd-cli/jd-cli.bat` [29 lines] *1. Resolve Java WITHOUT any hardcoded developer path. Prefer an ambient JAVA_HOME (manage.py pins it to the carried <install_dir>\jre in frozen. . .*
F1602 `Tlamatini/jd-cli/jd-cli.jar` [3.8MB] *jd-cli Java decompiler, run by the J-Decompiler agent and the decompile_java tool.*

agent/images/ · documentation media sheet 17

F0794 `Tlamatini/agent/images/ACPXKeysConfigureWizard.jpg` [118KB] *Image (JPG): ACPXKeysConfigureWizard.*
F0795 `Tlamatini/agent/images/ConfigureModels.jpg` [44KB] *Image (JPG): ConfigureModels.*
F0796 `Tlamatini/agent/images/MenuConfig.jpg` [18KB] *Image (JPG): MenuConfig.*
F0797 `Tlamatini/agent/images/mockup.jpg` [34KB] *Image (JPG): mockup.*
F0798 `Tlamatini/agent/images/XAIHT-Tlamatini.mp4` [10.8MB] *Tlamatini presentation video.*

agent/TlamatiniSourceCode/ · self-modify snapshot slot sheet 17

F1578 `Tlamatini/agent/TlamatiniSourceCode/README.md` [7 lines] *This folder contains the source code aimed to you in order to modify and improve yourself.*

agent/test_fixtures/ sheet 17

F1463 `Tlamatini/agent/test_fixtures/talker_orpheus_tokens.txt` [1 lines] *Recorded Orpheus audio-token stream (one line of custom tokens) replayed by the Talker tests without a model.*

Atlas IV — Presentation layer: first-party frontend

Torrenter workspace sheet 18

F1583 `Tlamatini/agent/torrenter_workspace/layout.js` [62 lines] *JavaScript module.*
F1584 `Tlamatini/agent/torrenter_workspace/workspace.css` [53 lines] *Stylesheet. [102 rule blocks]*
F1585 `Tlamatini/agent/torrenter_workspace/workspace.html` [28 lines] *Page "Torrenter · Tlachihuani".*
F1586 `Tlamatini/agent/torrenter_workspace/workspace.js` [212 lines] *Defines message(), api(), handle(), options() and 9 more.*

static/agent/interactor/ - tutor controls sheet 18

F0960 `Tlamatini/agent/static/agent/interactor/bridge.js` [305 lines] *Exports window.TlamatiniInteractor; 0 functions.*
F0961 `Tlamatini/agent/static/agent/interactor/interactor.css` [37 lines] *Stylesheet. [22 rule blocks]*

agent/static/agent/js/ · chat page (agent_page.html) sheet 18

F0962 `Tlamatini/agent/static/agent/js/access_keys_wizard.js` [406 lines] *access_keys_wizard.js - graphical one-page key setup wizard*
F0978 `Tlamatini/agent/static/agent/js/agent_page_canvas.js` [679 lines] *Shared classic-script bindings, consumed by sibling page scripts.*
F0979 `Tlamatini/agent/static/agent/js/agent_page_chat.js` [2.8k lines] *Shared classic-script bindings, consumed by sibling page scripts.*
F0980 `Tlamatini/agent/static/agent/js/agent_page_context.js` [188 lines] *Shared classic-script bindings, consumed by sibling page scripts.*
F0981 `Tlamatini/agent/static/agent/js/agent_page_dialogs.js` [1.4k lines] *Shared classic-script bindings, consumed by sibling page scripts.*
F0982 `Tlamatini/agent/static/agent/js/agent_page_editing.js` [308 lines] *File lifecycle stays in agent_page_canvas.js.*
F0983 `Tlamatini/agent/static/agent/js/agent_page_init.js` [1.9k lines] *agent_page_init.js – Initialization, event wiring & actions*
F0984 `Tlamatini/agent/static/agent/js/agent_page_layout.js` [469 lines] *Shared classic-script bindings, consumed by sibling page scripts.*
F0985 `Tlamatini/agent/static/agent/js/agent_page_markdown.js` [269 lines] *Sanitized Markdown presentation; edits always belong to the source editor.*
F0986 `Tlamatini/agent/static/agent/js/agent_page_pdf.js` [286 lines] *PDF canvas lifecycle and request bridge.*
F0987 `Tlamatini/agent/static/agent/js/agent_page_state.js` [495 lines] *Shared classic-script bindings, consumed by sibling page scripts.*
F0988 `Tlamatini/agent/static/agent/js/agent_page_ui.js` [521 lines] *Shared classic-script bindings, consumed by sibling page scripts.*
F0990 `Tlamatini/agent/static/agent/js/avatar.js` [437 lines] *Tlamatini chat-avatar engine - extracted from agent_page.html (2026-07-17).*
F0991 `Tlamatini/agent/static/agent/js/avatar_presence.js` [571 lines] *PRESENCE ENGINE — the talking head, composited on a 2D canvas. . . . WHAT THIS. . .*
F0992 `Tlamatini/agent/static/agent/js/avatar_size.js` [287 lines] *AVATAR SIZING — she is resizable, from 120 px up to most of the screen. . . .*
F0994 `Tlamatini/agent/static/agent/js/chat_dictation.js` [249 lines] *Direct Whisperer dictation; the existing chat form owns prompt submission.*
F0995 `Tlamatini/agent/static/agent/js/chat_image_paste.js` [390 lines] *chat_image_paste.js — screenshots into the chat box.*
F0996 `Tlamatini/agent/static/agent/js/chat_page_runtime_poller.js` [151 lines] *Polls /agent/check_chat_runtimes_status/ from the chat page (agent_page.html) and surfaces Asker A/B dialogs and Notifier toasts via. . .*
F0997 `Tlamatini/agent/static/agent/js/chat_table_contrast.js` [147 lines] *chat_table_contrast.js - every table in an answer stays READABLE (Angela Lopez Mendoza, 2026-10-01).*
F0999 `Tlamatini/agent/static/agent/js/compact_costs.js` [258 lines] *compact_costs.js — what each Configure row COSTS (Angela, 2026-10-02).*
F1000 `Tlamatini/agent/static/agent/js/contacts_dialog.js` [384 lines] *contacts_dialog.js — Config ▷ Contacts CRUD dialog*
F1001 `Tlamatini/agent/static/agent/js/context_gauge.js` [441 lines] *context_gauge.js — the context ring in the chat window (Angela López Mendoza, 2026-09-20 — Context Governor, step 1) Renders ONE row between the. . .*
F1004 `Tlamatini/agent/static/agent/js/external_mcps_dialog.js` [545 lines] *external_mcps_dialog.js — External ▷ MCPs menu*
F1009 `Tlamatini/agent/static/agent/js/mic_settings.js` [197 lines] *Browser preferences for direct dictation; microphone access stays in Whisperer.*
F1010 `Tlamatini/agent/static/agent/js/model_brain_tuning.js` [348 lines] *model_brain_tuning.js — the AUTO-TUNING dialog of the MODEL BRAIN (agent/model_brain.py), opened after Config ▷ Models ▷ Save.*
F1011 `Tlamatini/agent/static/agent/js/model_capacity.js` [492 lines] *model_capacity.js — the COMPACT MODE box, told to the user (Angela, 2026-10-01; the switch 2026-10-02).*
F1013 `Tlamatini/agent/static/agent/js/pdf_context_progress.js` [165 lines] *Accessible native modal; all behavior lives outside the HTML template.*
F1016 `Tlamatini/agent/static/agent/js/release_notes_renderer.js` [313 lines] *release_notes_renderer.js — renders GitHub RELEASE NOTES (Markdown + the raw HTML GitHub itself injects) into real HTML for the "About ▷ Check for. . .*
F1017 `Tlamatini/agent/static/agent/js/self_modify_switch.js` [282 lines] *self_modify_switch.js — the SELF-MODIFY box (Angela, 2026-10-03).*
F1019 `Tlamatini/agent/static/agent/js/skills_dialog.js` [392 lines] *skills_dialog.js — jQuery-UI dialogs for the ACPX-Skills menu*
F1021 `Tlamatini/agent/static/agent/js/tools_dialog.js` [688 lines] *Defines appendPrompt(), condenseWhitespace(), extractPromptTitle(), buildPromptPreview() and 12 more.*
F1023 `Tlamatini/agent/static/agent/js/usage_dashboard.js` [352 lines] *JavaScript module.*

agent/static/agent/js/ · Agentic Control Panel sheet 18

F0963 `Tlamatini/agent/static/agent/js/acp-agent-connectors.js` [1.7k lines] *Shared classic-script bindings, consumed by sibling page scripts.*
F0964 `Tlamatini/agent/static/agent/js/acp-canvas-core.js` [1.7k lines] *Shared classic-script bindings, consumed by sibling page scripts.*
F0965 `Tlamatini/agent/static/agent/js/acp-canvas-undo.js` [1.2k lines] *Agentic Control Panel - Canvas Undo/Redo Helpers & Keyboard Handler LOAD ORDER: #8 - Depends on: acp-globals.js, acp-session.js. . .*
F0966 `Tlamatini/agent/static/agent/js/acp-connection-status.js` [150 lines] *"BACKEND IS DOWN" BANNER FOR THE ACP DESIGNER*
F0967 `Tlamatini/agent/static/agent/js/acp-control-buttons.js` [1.4k lines] *Agentic Control Panel - Control Panel Button Handlers (Start / Stop / Pause / Clear) LOAD ORDER: #6 - Depends on: acp-globals.js, acp-session.js. . .*
F0968 `Tlamatini/agent/static/agent/js/acp-editor-tools.js` [319 lines] *Loaded after the existing canvas modules: all editing goes through their APIs.*
F0969 `Tlamatini/agent/static/agent/js/acp-file-io.js` [724 lines] *Agentic Control Panel - File I/O: Save, Open, Close, Load Diagram LOAD ORDER: #9 - Depends on: acp-globals.js, acp-session.js, acp-canvas-core.js. . .*
F0970 `Tlamatini/agent/static/agent/js/acp-flow-snapshot.js` [94 lines] *Agentic Control Panel - Flow Snapshot / Contract Compiler bridge LOAD ORDER: after acp-canvas-core.js; runtime dependency on getHeaders().*
F0971 `Tlamatini/agent/static/agent/js/acp-globals.js` [369 lines] *Shared classic-script bindings, consumed by sibling page scripts.*
F0972 `Tlamatini/agent/static/agent/js/acp-layout.js` [102 lines] *Agentic Control Panel - Layout, Divider, Agent List, Canvas Initialization LOAD ORDER: #10 - Depends on: ALL previous acp-*.js files*
F0973 `Tlamatini/agent/static/agent/js/acp-parametrizer-dialog.js` [563 lines] *Agentic Control Panel - Parametrizer Mapping Dialog LOAD ORDER: #11 - Depends on: acp-globals.js, acp-session.js*
F0974 `Tlamatini/agent/static/agent/js/acp-running-state.js` [370 lines] *Agentic Control Panel - Global Running State & Agent Status Polling LOAD ORDER: #5 - Depends on: acp-globals.js, acp-session.js, acp-undo-manager.js*
F0975 `Tlamatini/agent/static/agent/js/acp-session.js` [114 lines] *Agentic Control Panel - Session Management LOAD ORDER: #2 - Depends on: nothing (standalone)*
F0976 `Tlamatini/agent/static/agent/js/acp-undo-manager.js` [128 lines] *Shared classic-script bindings, consumed by sibling page scripts.*
F0977 `Tlamatini/agent/static/agent/js/acp-validate.js` [644 lines] *Shared classic-script bindings, consumed by sibling page scripts.*
F0993 `Tlamatini/agent/static/agent/js/canvas_item_dialog.js` [1.1k lines] *Defines preRenderCanvasItemDialog(), renderCanvasItemDialog(), showDeploymentResultDialog(), _executeFlowCreator() and 6 more.*
F1002 `Tlamatini/agent/static/agent/js/contextual_menus.js` [730 lines] *Contextual Menus for Agentic Control Panel Handles right-click context menus and log viewer modal for canvas agents*

agent/static/agent/js/ · Prompt Flow Panel sheet 18

F1014 `Tlamatini/agent/static/agent/js/prompt-flow-panel-model.js` [200 lines] *Exports window.PromptFlowPanelModel; 0 functions.*
F1015 `Tlamatini/agent/static/agent/js/prompt-flow-panel.js` [1.3k lines] *JavaScript module.*

agent/static/agent/js/ · shared by several pages sheet 18

F0998 `Tlamatini/agent/static/agent/js/checkbox_bulk_toggle.js` [319 lines] *checkbox_bulk_toggle.js – SPACE-bar bulk check/uncheck of every checkbox the current text selection covers.*
F1003 `Tlamatini/agent/static/agent/js/dialog_policy.js` [781 lines] *DIALOG DISMISSAL POLICY (Angela, 2026-08-13; ESCAPE STANDARDISED 2026-08-16) ONE rule, enforced in ONE place, for EVERY dialog on EVERY page: A. . .*
F1005 `Tlamatini/agent/static/agent/js/file_drop_router.js` [876 lines] *file_drop_router.js — ONE smart listener for every file dropped on ANY Tlamatini page.*
F1006 `Tlamatini/agent/static/agent/js/flow-canvas-interactions.js` [146 lines] *Exports window.FlowCanvasInteractions; 0 functions.*
F1007 `Tlamatini/agent/static/agent/js/interaction_settings.js` [232 lines] *Exports window.TlamatiniInteractionSettings; 0 functions.*
F1008 `Tlamatini/agent/static/agent/js/interaction_speech.js` [67 lines] *Exports window.TlamatiniInteractionSpeech; 0 functions.*
F1018 `Tlamatini/agent/static/agent/js/shared-runtime-dialogs.js` [347 lines] *Reusable Asker A/B dialog + Notifier toast renderers used by both agent_page.html (chat) and agentic_control_panel.html (ACP).*
F1020 `Tlamatini/agent/static/agent/js/terminater_dock.js` [952 lines] *Terminater dock — the Terminals area.*
F1022 `Tlamatini/agent/static/agent/js/ui_splitters.js` [62 lines] *Exports window.TlmSplitters; 0 functions.*
F1024 `Tlamatini/agent/static/agent/js/voice_runtime.js` [103 lines] *Shared speech lifecycle for chat, editor tutoring and the two flow panels.*

agent/static/agent/js/ · entry pages, PDF frame and indirect loads sheet 18

F0989 `Tlamatini/agent/static/agent/js/agentic_control_panel.js` [28 lines] *Agentic Control Panel - LEGACY STUB This file has been split into smaller functional modules.*
F1012 `Tlamatini/agent/static/agent/js/pdf_canvas_viewer.js` [213 lines] *Offline PDF.js viewer. Loaded only inside agent/pdf/canvas.html.*
F1025 `Tlamatini/agent/static/agent/js/welcome_enter_default.js` [92 lines] *welcome_enter_default.js — ENTER is the default action on the welcome page. welcome.html renders → press Enter → you are in the chat.*

agent/static/agent/css/ · chat page (agent_page.html) sheet 18

F0925 `Tlamatini/agent/static/agent/css/access_keys_wizard.css` [371 lines] *Graphical Access Keys Wizard [50 rule blocks]*
F0927 `Tlamatini/agent/static/agent/css/agent_page.css` [2.5k lines] *Stylesheet. [380 rule blocks]*
F0929 `Tlamatini/agent/static/agent/css/avatar.css` [181 lines] *Tlamatini chat-avatar - talking-head (blink + lip-sync), voice, bubble. [65 rule blocks]*
F0930 `Tlamatini/agent/static/agent/css/canvas_editor.css` [126 lines] *Stylesheet. [22 rule blocks]*
F0931 `Tlamatini/agent/static/agent/css/chat_dictation.css` [129 lines] *Stylesheet. [67 rule blocks]*
F0932 `Tlamatini/agent/static/agent/css/contacts_dialog.css` [538 lines] *contacts_dialog.css - Config > Contacts CRUD dialog Mirrors the External > MCPs native modal identity (emx-*) so the Contacts book feels native to. . . [69 rule blocks]*
F0933 `Tlamatini/agent/static/agent/css/context_gauge.css` [264 lines] *Context gauge — the ring in the chat window (Angela, 2026-09-20) A sibling row between the toolbar (#tools-div) and the message box, showing the. . . [36 rule blocks]*
F0936 `Tlamatini/agent/static/agent/css/external_mcps_dialog.css` [689 lines] *external_mcps_dialog.css - External > MCPs activation dialog Keeps the existing Tlamatini dark dialog identity while making the shell. . . [95 rule blocks]*
F0940 `Tlamatini/agent/static/agent/css/markdown_canvas.css` [61 lines] *Stylesheet. [42 rule blocks]*
F0941 `Tlamatini/agent/static/agent/css/model_brain_tuning.css` [327 lines] *model_brain_tuning.css — the Auto-tuning dialog of the MODEL BRAIN (Config ▷ Models ▷ Save). [62 rule blocks]*
F0942 `Tlamatini/agent/static/agent/css/model_capacity.css` [313 lines] *model_capacity.css — the Compact-mode badge, dialog body and notices (Angela, 2026-10-01). [48 rule blocks]*
F0944 `Tlamatini/agent/static/agent/css/pdf_context_progress.css` [27 lines] *Stylesheet. [21 rule blocks]*
F0946 `Tlamatini/agent/static/agent/css/skills_dialog.css` [337 lines] *skills_dialog.css — ACPX-Skills admin dialogs Three dialogs sharing one stylesheet: #skills-configure-dialog-message (checkbox grid; mirrors. . . [55 rule blocks]*
F0948 `Tlamatini/agent/static/agent/css/tools_dialog.css` [559 lines] *Stylesheet. [68 rule blocks]*
F0952 `Tlamatini/agent/static/agent/css/usage_dashboard.css` [114 lines] *Stylesheet. [123 rule blocks]*

agent/static/agent/css/ · Prompt Flow Panel sheet 18

F0945 `Tlamatini/agent/static/agent/css/prompt_flow_panel.css` [183 lines] *Stylesheet. [164 rule blocks]*

agent/static/agent/css/ · shared by several pages sheet 18

F0928 `Tlamatini/agent/static/agent/css/agentic_control_panel.css` [2.9k lines] *Existing styles [431 rule blocks]*
F0934 `Tlamatini/agent/static/agent/css/dialog_theme.css` [795 lines] *dialog_theme.css — THE ONE Tlamatini dialog identity Angela, 2026-08-12: every dialog in Tlamatini must look like the SAME piece of software. [129 rule blocks]*
F0937 `Tlamatini/agent/static/agent/css/flow_canvas.css` [58 lines] *Loaded after panel styles in both editors: common interaction states have one owner. [26 rule blocks]*
F0938 `Tlamatini/agent/static/agent/css/interaction_settings.css` [38 lines] *Stylesheet. [42 rule blocks]*
F0947 `Tlamatini/agent/static/agent/css/terminater_dock.css` [269 lines] *Terminater dock: the Terminals area (tabs, type bar, panes, tiling, overlays). [119 rule blocks]*
F0949 `Tlamatini/agent/static/agent/css/ui_shell.css` [182 lines] *Page chrome only. Diagram coordinates, content colors and dialog semantics belong to their existing owners. Load after page CSS, before dialog_theme. [90 rule blocks]*
F0950 `Tlamatini/agent/static/agent/css/ui_splitters.css` [39 lines] *One seam/grip treatment for every adjacent workspace pane. [22 rule blocks]*
F0951 `Tlamatini/agent/static/agent/css/ui_tokens.css` [30 lines] *Shared by the workspace, entry pages, dialogs, admin and isolated PDF frame. [1 rule blocks]*

agent/static/agent/css/ · entry pages, PDF frame and indirect loads sheet 18

F0926 `Tlamatini/agent/static/agent/css/admin_theme.css` [41 lines] *Keep Django's native light/dark/auto switch and accessible admin mechanics. [17 rule blocks]*
F0935 `Tlamatini/agent/static/agent/css/entry_pages.css` [38 lines] *Stylesheet. [23 rule blocks]*
F0939 `Tlamatini/agent/static/agent/css/login.css` [14 lines] *Stylesheet. [2 rule blocks]*
F0943 `Tlamatini/agent/static/agent/css/pdf_canvas.css` [19 lines] *The PDF renderer lives in its own frame so PDF.js styles cannot affect chat. [17 rule blocks]*
F0953 `Tlamatini/agent/static/agent/css/welcome.css` [13 lines] *Stylesheet. [1 rule blocks]*

agent/static/ · images, avatar, PDF frame, sounds, video sheet 18

F0954 `Tlamatini/agent/static/agent/img/avatar/ec_mc.jpg` [540KB] *Avatar portrait: eyes closed, mouth closed (blended by the presence compositor).*
F0955 `Tlamatini/agent/static/agent/img/avatar/ec_mo.jpg` [543KB] *Avatar portrait: eyes closed, mouth open (blended by the presence compositor).*
F0956 `Tlamatini/agent/static/agent/img/avatar/eo_mc.jpg` [534KB] *Avatar portrait: eyes open, mouth closed (blended by the presence compositor).*
F0957 `Tlamatini/agent/static/agent/img/avatar/eo_mo.jpg` [537KB] *Avatar portrait: eyes open, mouth open (blended by the presence compositor).*
F0958 `Tlamatini/agent/static/agent/img/spinner.svg` [18 lines] *SVG graphic: spinner.*
F0959 `Tlamatini/agent/static/agent/img/Tlamatini.ico` [38KB] *Tlamatini application icon.*
F1026 `Tlamatini/agent/static/agent/pdf/canvas.html` [52 lines] *Page "PDF canvas".*
F1027 `Tlamatini/agent/static/agent/sounds/hypervisor_alert.wav` [215KB] *Notification sound played by the chat/canvas pages.*
F1028 `Tlamatini/agent/static/agent/sounds/notification.wav` [279KB] *Notification sound played by the chat/canvas pages.*
F1379 `Tlamatini/agent/static/agent/video/XAIHT-Tlamatini.mp4` [10.8MB] *Tlamatini presentation video.*

Atlas V — Vendored third-party frontend and PDF.js

vendor/frontend sheet 19

F1062 `Tlamatini/agent/static/agent/vendor/frontend/manifest.json` [315 lines] *JSON document with keys: schema, packages, files.*

vendor/frontend/bootstrap sheet 19

F1032 `Tlamatini/agent/static/agent/vendor/frontend/bootstrap/dist/css/bootstrap.min.css` [6 lines] *Vendored Bootstrap 5 (navbar, dropdowns, modals, grid): bootstrap.min.css.*
F1033 `Tlamatini/agent/static/agent/vendor/frontend/bootstrap/dist/css/bootstrap.min.css.map` [1 lines] *Source map of the vendored minified bundle.*
F1034 `Tlamatini/agent/static/agent/vendor/frontend/bootstrap/dist/js/bootstrap.bundle.min.js` [7 lines] *Vendored Bootstrap 5 (navbar, dropdowns, modals, grid): bootstrap.bundle.min.js.*
F1035 `Tlamatini/agent/static/agent/vendor/frontend/bootstrap/dist/js/bootstrap.bundle.min.js.map` [1 lines] *Source map of the vendored minified bundle.*
F1036 `Tlamatini/agent/static/agent/vendor/frontend/bootstrap/LICENSE` [21 lines] *Third-party license text: The MIT License (MIT)*

vendor/frontend/dompurify sheet 19

F1037 `Tlamatini/agent/static/agent/vendor/frontend/dompurify/dist/purify.min.js` [7 lines] *Vendored DOMPurify HTML sanitizer used by the Markdown canvas: purify.min.js.*
F1038 `Tlamatini/agent/static/agent/vendor/frontend/dompurify/dist/purify.min.js.map` [1 lines] *Source map of the vendored minified bundle.*
F1039 `Tlamatini/agent/static/agent/vendor/frontend/dompurify/LICENSE` [202 lines] *Third-party license text: Apache License*

vendor/frontend/highlightjs sheet 19

F1040 `Tlamatini/agent/static/agent/vendor/frontend/highlightjs/highlight.min.js` [1.2k lines] *Vendored highlight.js syntax highlighter, languages and theme: highlight.min.js.*
F1041 `Tlamatini/agent/static/agent/vendor/frontend/highlightjs/languages/bash.min.js` [20 lines] *Vendored highlight.js syntax highlighter, languages and theme: bash.min.js.*
F1042 `Tlamatini/agent/static/agent/vendor/frontend/highlightjs/languages/c.min.js` [40 lines] *Vendored highlight.js syntax highlighter, languages and theme: c.min.js.*
F1043 `Tlamatini/agent/static/agent/vendor/frontend/highlightjs/languages/cpp.min.js` [47 lines] *Vendored highlight.js syntax highlighter, languages and theme: cpp.min.js.*
F1044 `Tlamatini/agent/static/agent/vendor/frontend/highlightjs/languages/css.min.js` [31 lines] *Vendored highlight.js syntax highlighter, languages and theme: css.min.js.*
F1045 `Tlamatini/agent/static/agent/vendor/frontend/highlightjs/languages/ini.min.js` [15 lines] *Vendored highlight.js syntax highlighter, languages and theme: ini.min.js.*
F1046 `Tlamatini/agent/static/agent/vendor/frontend/highlightjs/languages/java.min.js` [38 lines] *Vendored highlight.js syntax highlighter, languages and theme: java.min.js.*
F1047 `Tlamatini/agent/static/agent/vendor/frontend/highlightjs/languages/javascript.min.js` [80 lines] *Vendored highlight.js syntax highlighter, languages and theme: javascript.min.js.*
F1048 `Tlamatini/agent/static/agent/vendor/frontend/highlightjs/languages/kotlin.min.js` [46 lines] *Vendored highlight.js syntax highlighter, languages and theme: kotlin.min.js.*
F1049 `Tlamatini/agent/static/agent/vendor/frontend/highlightjs/languages/python.min.js` [41 lines] *Vendored highlight.js syntax highlighter, languages and theme: python.min.js.*
F1050 `Tlamatini/agent/static/agent/vendor/frontend/highlightjs/languages/rust.min.js` [28 lines] *Vendored highlight.js syntax highlighter, languages and theme: rust.min.js.*
F1051 `Tlamatini/agent/static/agent/vendor/frontend/highlightjs/languages/sql.min.js` [17 lines] *Vendored highlight.js syntax highlighter, languages and theme: sql.min.js.*
F1052 `Tlamatini/agent/static/agent/vendor/frontend/highlightjs/languages/typescript.min.js` [95 lines] *Vendored highlight.js syntax highlighter, languages and theme: typescript.min.js.*
F1053 `Tlamatini/agent/static/agent/vendor/frontend/highlightjs/languages/xml.min.js` [29 lines] *Vendored highlight.js syntax highlighter, languages and theme: xml.min.js.*
F1054 `Tlamatini/agent/static/agent/vendor/frontend/highlightjs/languages/yaml.min.js` [25 lines] *Vendored highlight.js syntax highlighter, languages and theme: yaml.min.js.*
F1055 `Tlamatini/agent/static/agent/vendor/frontend/highlightjs/LICENSE` [29 lines] *Third-party license text: BSD 3-Clause License*
F1056 `Tlamatini/agent/static/agent/vendor/frontend/highlightjs/styles/atom-one-dark.min.css` [1 lines] *Vendored highlight.js syntax highlighter, languages and theme: atom-one-dark.min.css.*

vendor/frontend/jquery sheet 19

F1059 `Tlamatini/agent/static/agent/vendor/frontend/jquery/dist/jquery.min.js` [2 lines] *Vendored jQuery: jquery.min.js.*
F1060 `Tlamatini/agent/static/agent/vendor/frontend/jquery/dist/jquery.min.map` [1 lines] *Source map of the vendored minified bundle.*
F1061 `Tlamatini/agent/static/agent/vendor/frontend/jquery/LICENSE.txt` [20 lines] *Third-party license text: Copyright OpenJS Foundation and other contributors, https://openjsf.org/*

vendor/frontend/jquery-ui sheet 19

F1057 `Tlamatini/agent/static/agent/vendor/frontend/jquery-ui/jquery-ui.min.js` [6 lines] *Vendored jQuery UI (dialogs, resizing): jquery-ui.min.js.*
F1058 `Tlamatini/agent/static/agent/vendor/frontend/jquery-ui/LICENSE.txt` [43 lines] *Third-party license text: Copyright OpenJS Foundation and other contributors, https://openjsf.org/*

vendor/frontend/marked sheet 19

F1063 `Tlamatini/agent/static/agent/vendor/frontend/marked/LICENSE` [44 lines] *Third-party license text: License information*

vendor/frontend/nunito sheet 19

F1064 `Tlamatini/agent/static/agent/vendor/frontend/nunito/400.css` [49 lines] *Vendored Nunito web font (site typography): 400.css.*
F1065 `Tlamatini/agent/static/agent/vendor/frontend/nunito/700.css` [49 lines] *Vendored Nunito web font (site typography): 700.css.*
F1066 `Tlamatini/agent/static/agent/vendor/frontend/nunito/files/nunito-cyrillic-400-normal.woff` [10KB] *Nunito web-font file (cyrillic-400-normal).*
F1067 `Tlamatini/agent/static/agent/vendor/frontend/nunito/files/nunito-cyrillic-400-normal.woff2` [8KB] *Nunito web-font file (cyrillic-400-normal).*
F1068 `Tlamatini/agent/static/agent/vendor/frontend/nunito/files/nunito-cyrillic-700-normal.woff` [10KB] *Nunito web-font file (cyrillic-700-normal).*
F1069 `Tlamatini/agent/static/agent/vendor/frontend/nunito/files/nunito-cyrillic-700-normal.woff2` [8KB] *Nunito web-font file (cyrillic-700-normal).*
F1070 `Tlamatini/agent/static/agent/vendor/frontend/nunito/files/nunito-cyrillic-ext-400-normal.woff` [14KB] *Nunito web-font file (cyrillic-ext-400-normal).*
F1071 `Tlamatini/agent/static/agent/vendor/frontend/nunito/files/nunito-cyrillic-ext-400-normal.woff2` [11KB] *Nunito web-font file (cyrillic-ext-400-normal).*

F1072 Tlamatini/agent/static/agent/vendor/frontend/nunito/files/nunito-cyrillic-ext-700-normal.woff [14KB] *Nunito web-font file (cyrillic-ext-700-normal).*
F1073 Tlamatini/agent/static/agent/vendor/frontend/nunito/files/nunito-cyrillic-ext-700-normal.woff2 [11KB] *Nunito web-font file (cyrillic-ext-700-normal).*
F1074 Tlamatini/agent/static/agent/vendor/frontend/nunito/files/nunito-latin-400-normal.woff [20KB] *Nunito web-font file (latin-400-normal).*
F1075 Tlamatini/agent/static/agent/vendor/frontend/nunito/files/nunito-latin-400-normal.woff2 [16KB] *Nunito web-font file (latin-400-normal).*
F1076 Tlamatini/agent/static/agent/vendor/frontend/nunito/files/nunito-latin-700-normal.woff [20KB] *Nunito web-font file (latin-700-normal).*
F1077 Tlamatini/agent/static/agent/vendor/frontend/nunito/files/nunito-latin-700-normal.woff2 [16KB] *Nunito web-font file (latin-700-normal).*
F1078 Tlamatini/agent/static/agent/vendor/frontend/nunito/files/nunito-latin-ext-400-normal.woff [21KB] *Nunito web-font file (latin-ext-400-normal).*
F1079 Tlamatini/agent/static/agent/vendor/frontend/nunito/files/nunito-latin-ext-400-normal.woff2 [16KB] *Nunito web-font file (latin-ext-400-normal).*
F1080 Tlamatini/agent/static/agent/vendor/frontend/nunito/files/nunito-latin-ext-700-normal.woff [21KB] *Nunito web-font file (latin-ext-700-normal).*
F1081 Tlamatini/agent/static/agent/vendor/frontend/nunito/files/nunito-latin-ext-700-normal.woff2 [16KB] *Nunito web-font file (latin-ext-700-normal).*
F1082 Tlamatini/agent/static/agent/vendor/frontend/nunito/files/nunito-vietnamese-400-normal.woff [8KB] *Nunito web-font file (vietnamese-400-normal).*
F1083 Tlamatini/agent/static/agent/vendor/frontend/nunito/files/nunito-vietnamese-400-normal.woff2 [6KB] *Nunito web-font file (vietnamese-400-normal).*
F1084 Tlamatini/agent/static/agent/vendor/frontend/nunito/files/nunito-vietnamese-700-normal.woff [8KB] *Nunito web-font file (vietnamese-700-normal).*
F1085 Tlamatini/agent/static/agent/vendor/frontend/nunito/files/nunito-vietnamese-700-normal.woff2 [6KB] *Nunito web-font file (vietnamese-700-normal).*
F1086 Tlamatini/agent/static/agent/vendor/frontend/nunito/LICENSE [93 lines] *Third-party license text: Google Inc.*

vendor/frontend/xterm — sheet 19

F1089 Tlamatini/agent/static/agent/vendor/frontend/xterm/css/xterm.css [218 lines] *Vendored xterm.js 5.5.0 terminal renderer of the Terminals area (Terminater): xterm.css.*
F1090 Tlamatini/agent/static/agent/vendor/frontend/xterm/LICENSE [21 lines] *Third-party license text: Copyright (c) 2017-2019, The xterm.js authors (https://github.com/xtermjs/xterm.js)*

vendor/frontend/xterm-addon-fit — sheet 19

F1087 Tlamatini/agent/static/agent/vendor/frontend/xterm-addon-fit/LICENSE [19 lines] *Third-party license text: Copyright (c) 2019, The xterm.js authors (https://github.com/xtermjs/xterm.js)*

vendor/frontend/xterm-addon-web-links — sheet 19

F1088 Tlamatini/agent/static/agent/vendor/frontend/xterm-addon-web-links/LICENSE [19 lines] *Third-party license text: Copyright (c) 2017, The xterm.js authors (https://github.com/xtermjs/xterm.js)*

vendor/canvas-editor - offline CodeMirror — sheet 19

F1029 Tlamatini/agent/static/agent/vendor/canvas-editor/editor.js [35 lines] *Vendored canvas-editor: editor.js.*
F1030 Tlamatini/agent/static/agent/vendor/canvas-editor/LICENSES.txt [953 lines] *Third-party license text: @codemirror/autocomplete 6.20.4*
F1031 Tlamatini/agent/static/agent/vendor/canvas-editor/manifest.json [50 lines] *JSON document with keys: source, build, bytes, sha256, lock_file, lock_sha256, packages.*

vendor/pdfjs — sheet 19

F1264 Tlamatini/agent/static/agent/vendor/pdfjs/LICENSE [177 lines] *Third-party license text: Apache License*
F1281 Tlamatini/agent/static/agent/vendor/pdfjs/VERSION [1 lines] *Pinned PDF.js version: 6.3.289.*

vendor/pdfjs/build — sheet 19

F1091 Tlamatini/agent/static/agent/vendor/pdfjs/build/pdf.mjs [28k lines] *Vendored PDF.js core library and worker: pdf.mjs.*
F1092 Tlamatini/agent/static/agent/vendor/pdfjs/build/pdf.worker.mjs [65k lines] *Vendored PDF.js core library and worker: pdf.worker.mjs.*

vendor/pdfjs/cmaps — sheet 19

F1093 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/78-EUC-H.bcmap [2KB] *PDF.js binary CMap 78-EUC-H (character-code to CID mapping for the PDF canvas).*
F1094 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/78-EUC-V.bcmap [173B] *PDF.js binary CMap 78-EUC-V (character-code to CID mapping for the PDF canvas).*
F1095 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/78-H.bcmap [2KB] *PDF.js binary CMap 78-H (character-code to CID mapping for the PDF canvas).*
F1096 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/78-RKSJ-H.bcmap [2KB] *PDF.js binary CMap 78-RKSJ-H (character-code to CID mapping for the PDF canvas).*
F1097 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/78-RKSJ-V.bcmap [173B] *PDF.js binary CMap 78-RKSJ-V (character-code to CID mapping for the PDF canvas).*
F1098 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/78-V.bcmap [169B] *PDF.js binary CMap 78-V (character-code to CID mapping for the PDF canvas).*
F1099 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/78ms-RKSJ-H.bcmap [3KB] *PDF.js binary CMap 78ms-RKSJ-H (character-code to CID mapping for the PDF canvas).*
F1100 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/78ms-RKSJ-V.bcmap [290B] *PDF.js binary CMap 78ms-RKSJ-V (character-code to CID mapping for the PDF canvas).*
F1101 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/83pv-RKSJ-H.bcmap [905B] *PDF.js binary CMap 83pv-RKSJ-H (character-code to CID mapping for the PDF canvas).*
F1102 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/90ms-RKSJ-H.bcmap [721B] *PDF.js binary CMap 90ms-RKSJ-H (character-code to CID mapping for the PDF canvas).*
F1103 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/90ms-RKSJ-V.bcmap [290B] *PDF.js binary CMap 90ms-RKSJ-V (character-code to CID mapping for the PDF canvas).*
F1104 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/90msp-RKSJ-H.bcmap [715B] *PDF.js binary CMap 90msp-RKSJ-H (character-code to CID mapping for the PDF canvas).*
F1105 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/90msp-RKSJ-V.bcmap [291B] *PDF.js binary CMap 90msp-RKSJ-V (character-code to CID mapping for the PDF canvas).*
F1106 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/90pv-RKSJ-H.bcmap [982B] *PDF.js binary CMap 90pv-RKSJ-H (character-code to CID mapping for the PDF canvas).*
F1107 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/90pv-RKSJ-V.bcmap [260B] *PDF.js binary CMap 90pv-RKSJ-V (character-code to CID mapping for the PDF canvas).*
F1108 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Add-H.bcmap [2KB] *PDF.js binary CMap Add-H (character-code to CID mapping for the PDF canvas).*
F1109 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Add-RKSJ-H.bcmap [2KB] *PDF.js binary CMap Add-RKSJ-H (character-code to CID mapping for the PDF canvas).*
F1110 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Add-RKSJ-V.bcmap [287B] *PDF.js binary CMap Add-RKSJ-V (character-code to CID mapping for the PDF canvas).*
F1111 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Add-V.bcmap [282B] *PDF.js binary CMap Add-V (character-code to CID mapping for the PDF canvas).*
F1112 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-CNS1-0.bcmap [317B] *PDF.js binary CMap Adobe-CNS1-0 (character-code to CID mapping for the PDF canvas).*
F1113 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-CNS1-1.bcmap [371B] *PDF.js binary CMap Adobe-CNS1-1 (character-code to CID mapping for the PDF canvas).*
F1114 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-CNS1-2.bcmap [376B] *PDF.js binary CMap Adobe-CNS1-2 (character-code to CID mapping for the PDF canvas).*
F1115 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-CNS1-3.bcmap [401B] *PDF.js binary CMap Adobe-CNS1-3 (character-code to CID mapping for the PDF canvas).*
F1116 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-CNS1-4.bcmap [405B] *PDF.js binary CMap Adobe-CNS1-4 (character-code to CID mapping for the PDF canvas).*
F1117 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-CNS1-5.bcmap [406B] *PDF.js binary CMap Adobe-CNS1-5 (character-code to CID mapping for the PDF canvas).*
F1118 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-CNS1-6.bcmap [406B] *PDF.js binary CMap Adobe-CNS1-6 (character-code to CID mapping for the PDF canvas).*
F1119 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-CNS1-UCS2.bcmap [40KB] *PDF.js binary CMap Adobe-CNS1-UCS2 (character-code to CID mapping for the PDF canvas).*
F1120 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-GB1-0.bcmap [217B] *PDF.js binary CMap Adobe-GB1-0 (character-code to CID mapping for the PDF canvas).*
F1121 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-GB1-1.bcmap [250B] *PDF.js binary CMap Adobe-GB1-1 (character-code to CID mapping for the PDF canvas).*
F1122 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-GB1-2.bcmap [465B] *PDF.js binary CMap Adobe-GB1-2 (character-code to CID mapping for the PDF canvas).*
F1123 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-GB1-3.bcmap [470B] *PDF.js binary CMap Adobe-GB1-3 (character-code to CID mapping for the PDF canvas).*
F1124 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-GB1-4.bcmap [601B] *PDF.js binary CMap Adobe-GB1-4 (character-code to CID mapping for the PDF canvas).*
F1125 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-GB1-5.bcmap [625B] *PDF.js binary CMap Adobe-GB1-5 (character-code to CID mapping for the PDF canvas).*
F1126 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-GB1-UCS2.bcmap [33KB] *PDF.js binary CMap Adobe-GB1-UCS2 (character-code to CID mapping for the PDF canvas).*
F1127 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-Japan1-0.bcmap [225B] *PDF.js binary CMap Adobe-Japan1-0 (character-code to CID mapping for the PDF canvas).*
F1128 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-Japan1-1.bcmap [226B] *PDF.js binary CMap Adobe-Japan1-1 (character-code to CID mapping for the PDF canvas).*
F1129 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-Japan1-2.bcmap [233B] *PDF.js binary CMap Adobe-Japan1-2 (character-code to CID mapping for the PDF canvas).*
F1130 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-Japan1-3.bcmap [242B] *PDF.js binary CMap Adobe-Japan1-3 (character-code to CID mapping for the PDF canvas).*
F1131 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-Japan1-4.bcmap [337B] *PDF.js binary CMap Adobe-Japan1-4 (character-code to CID mapping for the PDF canvas).*
F1132 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-Japan1-5.bcmap [430B] *PDF.js binary CMap Adobe-Japan1-5 (character-code to CID mapping for the PDF canvas).*
F1133 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-Japan1-6.bcmap [485B] *PDF.js binary CMap Adobe-Japan1-6 (character-code to CID mapping for the PDF canvas).*
F1134 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-Japan1-UCS2.bcmap [40KB] *PDF.js binary CMap Adobe-Japan1-UCS2 (character-code to CID mapping for the PDF canvas).*
F1135 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-Korea1-0.bcmap [241B] *PDF.js binary CMap Adobe-Korea1-0 (character-code to CID mapping for the PDF canvas).*
F1136 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-Korea1-1.bcmap [386B] *PDF.js binary CMap Adobe-Korea1-1 (character-code to CID mapping for the PDF canvas).*
F1137 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-Korea1-2.bcmap [391B] *PDF.js binary CMap Adobe-Korea1-2 (character-code to CID mapping for the PDF canvas).*
F1138 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Adobe-Korea1-UCS2.bcmap [23KB] *PDF.js binary CMap Adobe-Korea1-UCS2 (character-code to CID mapping for the PDF canvas).*
F1139 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/B5-H.bcmap [1KB] *PDF.js binary CMap B5-H (character-code to CID mapping for the PDF canvas).*
F1140 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/B5-V.bcmap [142B] *PDF.js binary CMap B5-V (character-code to CID mapping for the PDF canvas).*
F1141 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/B5pc-H.bcmap [1KB] *PDF.js binary CMap B5pc-H (character-code to CID mapping for the PDF canvas).*
F1142 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/B5pc-V.bcmap [144B] *PDF.js binary CMap B5pc-V (character-code to CID mapping for the PDF canvas).*
F1143 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/CNS-EUC-H.bcmap [2KB] *PDF.js binary CMap CNS-EUC-H (character-code to CID mapping for the PDF canvas).*
F1144 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/CNS-EUC-V.bcmap [2KB] *PDF.js binary CMap CNS-EUC-V (character-code to CID mapping for the PDF canvas).*
F1145 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/CNS1-H.bcmap [706B] *PDF.js binary CMap CNS1-H (character-code to CID mapping for the PDF canvas).*
F1146 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/CNS1-V.bcmap [143B] *PDF.js binary CMap CNS1-V (character-code to CID mapping for the PDF canvas).*
F1147 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/CNS2-H.bcmap [504B] *PDF.js binary CMap CNS2-H (character-code to CID mapping for the PDF canvas).*
F1148 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/CNS2-V.bcmap [93B] *PDF.js binary CMap CNS2-V (character-code to CID mapping for the PDF canvas).*
F1149 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/ETen-B5-H.bcmap [1KB] *PDF.js binary CMap ETen-B5-H (character-code to CID mapping for the PDF canvas).*
F1150 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/ETen-B5-V.bcmap [158B] *PDF.js binary CMap ETen-B5-V (character-code to CID mapping for the PDF canvas).*
F1151 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/ETenms-B5-H.bcmap [101B] *PDF.js binary CMap ETenms-B5-H (character-code to CID mapping for the PDF canvas).*
F1152 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/ETenms-B5-V.bcmap [172B] *PDF.js binary CMap ETenms-B5-V (character-code to CID mapping for the PDF canvas).*
F1153 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/ETHK-B5-H.bcmap [4KB] *PDF.js binary CMap ETHK-B5-H (character-code to CID mapping for the PDF canvas).*
F1154 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/ETHK-B5-V.bcmap [158B] *PDF.js binary CMap ETHK-B5-V (character-code to CID mapping for the PDF canvas).*
F1155 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/EUC-H.bcmap [578B] *PDF.js binary CMap EUC-H (character-code to CID mapping for the PDF canvas).*
F1156 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/EUC-V.bcmap [170B] *PDF.js binary CMap EUC-V (character-code to CID mapping for the PDF canvas).*
F1157 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Ext-H.bcmap [2KB] *PDF.js binary CMap Ext-H (character-code to CID mapping for the PDF canvas).*
F1158 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Ext-RKSJ-H.bcmap [2KB] *PDF.js binary CMap Ext-RKSJ-H (character-code to CID mapping for the PDF canvas).*
F1159 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Ext-RKSJ-V.bcmap [218B] *PDF.js binary CMap Ext-RKSJ-V (character-code to CID mapping for the PDF canvas).*
F1160 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Ext-V.bcmap [215B] *PDF.js binary CMap Ext-V (character-code to CID mapping for the PDF canvas).*
F1161 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/GB-EUC-H.bcmap [549B] *PDF.js binary CMap GB-EUC-H (character-code to CID mapping for the PDF canvas).*
F1162 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/GB-EUC-V.bcmap [179B] *PDF.js binary CMap GB-EUC-V (character-code to CID mapping for the PDF canvas).*
F1163 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/GB-H.bcmap [528B] *PDF.js binary CMap GB-H (character-code to CID mapping for the PDF canvas).*
F1164 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/GB-V.bcmap [175B] *PDF.js binary CMap GB-V (character-code to CID mapping for the PDF canvas).*
F1165 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/GBK-EUC-H.bcmap [14KB] *PDF.js binary CMap GBK-EUC-H (character-code to CID mapping for the PDF canvas).*
F1166 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/GBK-EUC-V.bcmap [180B] *PDF.js binary CMap GBK-EUC-V (character-code to CID mapping for the PDF canvas).*
F1167 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/GBK2K-H.bcmap [19KB] *PDF.js binary CMap GBK2K-H (character-code to CID mapping for the PDF canvas).*
F1168 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/GBK2K-V.bcmap [219B] *PDF.js binary CMap GBK2K-V (character-code to CID mapping for the PDF canvas).*
F1169 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/GBKp-EUC-H.bcmap [14KB] *PDF.js binary CMap GBKp-EUC-H (character-code to CID mapping for the PDF canvas).*
F1170 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/GBKp-EUC-V.bcmap [181B] *PDF.js binary CMap GBKp-EUC-V (character-code to CID mapping for the PDF canvas).*
F1171 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/GBpc-EUC-H.bcmap [557B] *PDF.js binary CMap GBpc-EUC-H (character-code to CID mapping for the PDF canvas).*
F1172 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/GBpc-EUC-V.bcmap [181B] *PDF.js binary CMap GBpc-EUC-V (character-code to CID mapping for the PDF canvas).*
F1173 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/GBT-EUC-H.bcmap [7KB] *PDF.js binary CMap GBT-EUC-H (character-code to CID mapping for the PDF canvas).*
F1174 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/GBT-EUC-V.bcmap [180B] *PDF.js binary CMap GBT-EUC-V (character-code to CID mapping for the PDF canvas).*
F1175 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/GBT-H.bcmap [7KB] *PDF.js binary CMap GBT-H (character-code to CID mapping for the PDF canvas).*
F1176 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/GBT-V.bcmap [176B] *PDF.js binary CMap GBT-V (character-code to CID mapping for the PDF canvas).*
F1177 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/GBTpc-EUC-H.bcmap [7KB] *PDF.js binary CMap GBTpc-EUC-H (character-code to CID mapping for the PDF canvas).*
F1178 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/GBTpc-EUC-V.bcmap [182B] *PDF.js binary CMap GBTpc-EUC-V (character-code to CID mapping for the PDF canvas).*
F1179 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/H.bcmap [553B] *PDF.js binary CMap H (character-code to CID mapping for the PDF canvas).*
F1180 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Hankaku.bcmap [132B] *PDF.js binary CMap Hankaku (character-code to CID mapping for the PDF canvas).*
F1181 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Hiragana.bcmap [124B] *PDF.js binary CMap Hiragana (character-code to CID mapping for the PDF canvas).*
F1182 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/HKdla-B5-H.bcmap [3KB] *PDF.js binary CMap HKdla-B5-H (character-code to CID mapping for the PDF canvas).*
F1183 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/HKdla-B5-V.bcmap [148B] *PDF.js binary CMap HKdla-B5-V (character-code to CID mapping for the PDF canvas).*
F1184 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/HKdlb-B5-H.bcmap [2KB] *PDF.js binary CMap HKdlb-B5-H (character-code to CID mapping for the PDF canvas).*
F1185 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/HKdlb-B5-V.bcmap [148B] *PDF.js binary CMap HKdlb-B5-V (character-code to CID mapping for the PDF canvas).*
F1186 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/HKgccs-B5-H.bcmap [2KB] *PDF.js binary CMap HKgccs-B5-H (character-code to CID mapping for the PDF canvas).*
F1187 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/HKgccs-B5-V.bcmap [149B] *PDF.js binary CMap HKgccs-B5-V (character-code to CID mapping for the PDF canvas).*
F1188 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/HKm314-B5-H.bcmap [2KB] *PDF.js binary CMap HKm314-B5-H (character-code to CID mapping for the PDF canvas).*
F1189 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/HKm314-B5-V.bcmap [149B] *PDF.js binary CMap HKm314-B5-V (character-code to CID mapping for the PDF canvas).*
F1190 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/HKm471-B5-H.bcmap [2KB] *PDF.js binary CMap HKm471-B5-H (character-code to CID mapping for the PDF canvas).*
F1191 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/HKm471-B5-V.bcmap [149B] *PDF.js binary CMap HKm471-B5-V (character-code to CID mapping for the PDF canvas).*
F1192 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/HKscs-B5-H.bcmap [4KB] *PDF.js binary CMap HKscs-B5-H (character-code to CID mapping for the PDF canvas).*
F1193 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/HKscs-B5-V.bcmap [159B] *PDF.js binary CMap HKscs-B5-V (character-code to CID mapping for the PDF canvas).*
F1194 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Katakana.bcmap [100B] *PDF.js binary CMap Katakana (character-code to CID mapping for the PDF canvas).*
F1195 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/KSC-EUC-H.bcmap [2KB] *PDF.js binary CMap KSC-EUC-H (character-code to CID mapping for the PDF canvas).*
F1196 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/KSC-EUC-V.bcmap [164B] *PDF.js binary CMap KSC-EUC-V (character-code to CID mapping for the PDF canvas).*
F1197 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/KSC-H.bcmap [2KB] *PDF.js binary CMap KSC-H (character-code to CID mapping for the PDF canvas).*
F1198 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/KSC-Johab-H.bcmap [16KB] *PDF.js binary CMap KSC-Johab-H (character-code to CID mapping for the PDF canvas).*
F1199 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/KSC-Johab-V.bcmap [166B] *PDF.js binary CMap KSC-Johab-V (character-code to CID mapping for the PDF canvas).*
F1200 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/KSC-V.bcmap [160B] *PDF.js binary CMap KSC-V (character-code to CID mapping for the PDF canvas).*
F1201 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/KSCms-UHC-H.bcmap [3KB] *PDF.js binary CMap KSCms-UHC-H (character-code to CID mapping for the PDF canvas).*
F1202 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/KSCms-UHC-HW-H.bcmap [3KB] *PDF.js binary CMap KSCms-UHC-HW-H (character-code to CID mapping for the PDF canvas).*
F1203 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/KSCms-UHC-HW-V.bcmap [169B] *PDF.js binary CMap KSCms-UHC-HW-V (character-code to CID mapping for the PDF canvas).*
F1204 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/KSCms-UHC-V.bcmap [166B] *PDF.js binary CMap KSCms-UHC-V (character-code to CID mapping for the PDF canvas).*
F1205 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/KSCpc-EUC-H.bcmap [2KB] *PDF.js binary CMap KSCpc-EUC-H (character-code to CID mapping for the PDF canvas).*
F1206 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/KSCpc-EUC-V.bcmap [166B] *PDF.js binary CMap KSCpc-EUC-V (character-code to CID mapping for the PDF canvas).*
F1207 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/LICENSE [36 lines] *Third-party license text: %%Copyright*
F1208 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/NWP-H.bcmap [3KB] *PDF.js binary CMap NWP-H (character-code to CID mapping for the PDF canvas).*
F1209 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/NWP-V.bcmap [252B] *PDF.js binary CMap NWP-V (character-code to CID mapping for the PDF canvas).*
F1210 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/RKSJ-H.bcmap [534B] *PDF.js binary CMap RKSJ-H (character-code to CID mapping for the PDF canvas).*
F1211 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/RKSJ-V.bcmap [170B] *PDF.js binary CMap RKSJ-V (character-code to CID mapping for the PDF canvas).*
F1212 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/Roman.bcmap [96B] *PDF.js binary CMap Roman (character-code to CID mapping for the PDF canvas).*
F1213 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniCNS-UCS2-H.bcmap [47KB] *PDF.js binary CMap UniCNS-UCS2-H (character-code to CID mapping for the PDF canvas).*
F1214 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniCNS-UCS2-V.bcmap [156B] *PDF.js binary CMap UniCNS-UCS2-V (character-code to CID mapping for the PDF canvas).*
F1215 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniCNS-UTF16-H.bcmap [49KB] *PDF.js binary CMap UniCNS-UTF16-H (character-code to CID mapping for the PDF canvas).*
F1216 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniCNS-UTF16-V.bcmap [156B] *PDF.js binary CMap UniCNS-UTF16-V (character-code to CID mapping for the PDF canvas).*
F1217 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniCNS-UTF32-H.bcmap [51KB] *PDF.js binary CMap UniCNS-UTF32-H (character-code to CID mapping for the PDF canvas).*
F1218 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniCNS-UTF32-V.bcmap [160B] *PDF.js binary CMap UniCNS-UTF32-V (character-code to CID mapping for the PDF canvas).*
F1219 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniCNS-UTF8-H.bcmap [52KB] *PDF.js binary CMap UniCNS-UTF8-H (character-code to CID mapping for the PDF canvas).*
F1220 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniCNS-UTF8-V.bcmap [157B] *PDF.js binary CMap UniCNS-UTF8-V (character-code to CID mapping for the PDF canvas).*
F1221 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniGB-UCS2-H.bcmap [42KB] *PDF.js binary CMap UniGB-UCS2-H (character-code to CID mapping for the PDF canvas).*
F1222 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniGB-UCS2-V.bcmap [193B] *PDF.js binary CMap UniGB-UCS2-V (character-code to CID mapping for the PDF canvas).*
F1223 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniGB-UTF16-H.bcmap [43KB] *PDF.js binary CMap UniGB-UTF16-H (character-code to CID mapping for the PDF canvas).*
F1224 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniGB-UTF16-V.bcmap [178B] *PDF.js binary CMap UniGB-UTF16-V (character-code to CID mapping for the PDF canvas).*
F1225 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniGB-UTF32-H.bcmap [45KB] *PDF.js binary CMap UniGB-UTF32-H (character-code to CID mapping for the PDF canvas).*
F1226 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniGB-UTF32-V.bcmap [182B] *PDF.js binary CMap UniGB-UTF32-V (character-code to CID mapping for the PDF canvas).*
F1227 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniGB-UTF8-H.bcmap [46KB] *PDF.js binary CMap UniGB-UTF8-H (character-code to CID mapping for the PDF canvas).*
F1228 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniGB-UTF8-V.bcmap [181B] *PDF.js binary CMap UniGB-UTF8-V (character-code to CID mapping for the PDF canvas).*
F1229 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJIS-UCS2-H.bcmap [25KB] *PDF.js binary CMap UniJIS-UCS2-H (character-code to CID mapping for the PDF canvas).*
F1230 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJIS-UCS2-HW-H.bcmap [119B] *PDF.js binary CMap UniJIS-UCS2-HW-H (character-code to CID mapping for the PDF canvas).*
F1231 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJIS-UCS2-HW-V.bcmap [680B] *PDF.js binary CMap UniJIS-UCS2-HW-V (character-code to CID mapping for the PDF canvas).*
F1232 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJIS-UCS2-V.bcmap [664B] *PDF.js binary CMap UniJIS-UCS2-V (character-code to CID mapping for the PDF canvas).*
F1233 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJIS-UTF16-H.bcmap [39KB] *PDF.js binary CMap UniJIS-UTF16-H (character-code to CID mapping for the PDF canvas).*
F1234 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJIS-UTF16-V.bcmap [643B] *PDF.js binary CMap UniJIS-UTF16-V (character-code to CID mapping for the PDF canvas).*
F1235 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJIS-UTF32-H.bcmap [40KB] *PDF.js binary CMap UniJIS-UTF32-H (character-code to CID mapping for the PDF canvas).*
F1236 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJIS-UTF32-V.bcmap [677B] *PDF.js binary CMap UniJIS-UTF32-V (character-code to CID mapping for the PDF canvas).*
F1237 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJIS-UTF8-H.bcmap [41KB] *PDF.js binary CMap UniJIS-UTF8-H (character-code to CID mapping for the PDF canvas).*
F1238 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJIS-UTF8-V.bcmap [678B] *PDF.js binary CMap UniJIS-UTF8-V (character-code to CID mapping for the PDF canvas).*
F1239 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJIS2004-UTF16-H.bcmap [39KB] *PDF.js binary CMap UniJIS2004-UTF16-H (character-code to CID mapping for the PDF canvas).*
F1240 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJIS2004-UTF16-V.bcmap [647B] *PDF.js binary CMap UniJIS2004-UTF16-V (character-code to CID mapping for the PDF canvas).*
F1241 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJIS2004-UTF32-H.bcmap [40KB] *PDF.js binary CMap UniJIS2004-UTF32-H (character-code to CID mapping for the PDF canvas).*
F1242 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJIS2004-UTF32-V.bcmap [681B] *PDF.js binary CMap UniJIS2004-UTF32-V (character-code to CID mapping for the PDF canvas).*
F1243 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJIS2004-UTF8-H.bcmap [41KB] *PDF.js binary CMap UniJIS2004-UTF8-H (character-code to CID mapping for the PDF canvas).*
F1244 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJIS2004-UTF8-V.bcmap [682B] *PDF.js binary CMap UniJIS2004-UTF8-V (character-code to CID mapping for the PDF canvas).*
F1245 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJISPro-UCS2-HW-V.bcmap [705B] *PDF.js binary CMap UniJISPro-UCS2-HW-V (character-code to CID mapping for the PDF canvas).*
F1246 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJISPro-UCS2-V.bcmap [689B] *PDF.js binary CMap UniJISPro-UCS2-V (character-code to CID mapping for the PDF canvas).*
F1247 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJISPro-UTF8-V.bcmap [726B] *PDF.js binary CMap UniJISPro-UTF8-V (character-code to CID mapping for the PDF canvas).*
F1248 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJISX0213-UTF32-H.bcmap [40KB] *PDF.js binary CMap UniJISX0213-UTF32-H (character-code to CID mapping for the PDF canvas).*
F1249 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJISX0213-UTF32-V.bcmap [684B] *PDF.js binary CMap UniJISX0213-UTF32-V (character-code to CID mapping for the PDF canvas).*
F1250 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJISX02132004-UTF32-H.bcmap [40KB] *PDF.js binary CMap UniJISX02132004-UTF32-H (character-code to CID mapping for the PDF canvas).*
F1251 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniJISX02132004-UTF32-V.bcmap [688B] *PDF.js binary CMap UniJISX02132004-UTF32-V (character-code to CID mapping for the PDF canvas).*
F1252 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniKS-UCS2-H.bcmap [25KB] *PDF.js binary CMap UniKS-UCS2-H (character-code to CID mapping for the PDF canvas).*
F1253 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniKS-UCS2-V.bcmap [178B] *PDF.js binary CMap UniKS-UCS2-V (character-code to CID mapping for the PDF canvas).*
F1254 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniKS-UTF16-H.bcmap [26KB] *PDF.js binary CMap UniKS-UTF16-H (character-code to CID mapping for the PDF canvas).*
F1255 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniKS-UTF16-V.bcmap [164B] *PDF.js binary CMap UniKS-UTF16-V (character-code to CID mapping for the PDF canvas).*
F1256 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniKS-UTF32-H.bcmap [26KB] *PDF.js binary CMap UniKS-UTF32-H (character-code to CID mapping for the PDF canvas).*
F1257 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniKS-UTF32-V.bcmap [168B] *PDF.js binary CMap UniKS-UTF32-V (character-code to CID mapping for the PDF canvas).*
F1258 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniKS-UTF8-H.bcmap [27KB] *PDF.js binary CMap UniKS-UTF8-H (character-code to CID mapping for the PDF canvas).*
F1259 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/UniKS-UTF8-V.bcmap [169B] *PDF.js binary CMap UniKS-UTF8-V (character-code to CID mapping for the PDF canvas).*
F1260 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/V.bcmap [166B] *PDF.js binary CMap V (character-code to CID mapping for the PDF canvas).*
F1261 Tlamatini/agent/static/agent/vendor/pdfjs/cmaps/WP-Symbol.bcmap [179B] *PDF.js binary CMap WP-Symbol (character-code to CID mapping for the PDF canvas).*

vendor/pdfjs/iccs — sheet 19

F1262 Tlamatini/agent/static/agent/vendor/pdfjs/iccs/CGATS001Compat-v2-micro.icc [8KB] *ICC color profile used by PDF.js.*
F1263 Tlamatini/agent/static/agent/vendor/pdfjs/iccs/LICENSE [116 lines] *Third-party license text: CC0 1.0 Universal*

vendor/pdfjs/standard_fonts — sheet 19

F1265 Tlamatini/agent/static/agent/vendor/pdfjs/standard_fonts/FoxitDingbats.pfb [29KB] *Standard font shipped with PDF.js for PDFs that do not embed it.*
F1266 Tlamatini/agent/static/agent/vendor/pdfjs/standard_fonts/FoxitFixed.pfb [17KB] *Standard font shipped with PDF.js for PDFs that do not embed it.*
F1267 Tlamatini/agent/static/agent/vendor/pdfjs/standard_fonts/FoxitFixedBold.pfb [18KB] *Standard font shipped with PDF.js for PDFs that do not embed it.*
F1268 Tlamatini/agent/static/agent/vendor/pdfjs/standard_fonts/FoxitFixedBoldItalic.pfb [19KB] *Standard font shipped with PDF.js for PDFs that do not embed it.*
F1269 Tlamatini/agent/static/agent/vendor/pdfjs/standard_fonts/FoxitFixedItalic.pfb [18KB] *Standard font shipped with PDF.js for PDFs that do not embed it.*
F1270 Tlamatini/agent/static/agent/vendor/pdfjs/standard_fonts/FoxitSerif.pfb [19KB] *Standard font shipped with PDF.js for PDFs that do not embed it.*
F1271 Tlamatini/agent/static/agent/vendor/pdfjs/standard_fonts/FoxitSerifBold.pfb [19KB] *Standard font shipped with PDF.js for PDFs that do not embed it.*
F1272 Tlamatini/agent/static/agent/vendor/pdfjs/standard_fonts/FoxitSerifBoldItalic.pfb [20KB] *Standard font shipped with PDF.js for PDFs that do not embed it.*
F1273 Tlamatini/agent/static/agent/vendor/pdfjs/standard_fonts/FoxitSerifItalic.pfb [21KB] *Standard font shipped with PDF.js for PDFs that do not embed it.*
F1274 Tlamatini/agent/static/agent/vendor/pdfjs/standard_fonts/FoxitSymbol.pfb [16KB] *Standard font shipped with PDF.js for PDFs that do not embed it.*
F1275 Tlamatini/agent/static/agent/vendor/pdfjs/standard_fonts/LiberationSans-Bold.ttf [134KB] *Standard font shipped with PDF.js for PDFs that do not embed it.*
F1276 Tlamatini/agent/static/agent/vendor/pdfjs/standard_fonts/LiberationSans-BoldItalic.ttf [132KB] *Standard font shipped with PDF.js for PDFs that do not embed it.*
F1277 Tlamatini/agent/static/agent/vendor/pdfjs/standard_fonts/LiberationSans-Italic.ttf [158KB] *Standard font shipped with PDF.js for PDFs that do not embed it.*
F1278 Tlamatini/agent/static/agent/vendor/pdfjs/standard_fonts/LiberationSans-Regular.ttf [136KB] *Standard font shipped with PDF.js for PDFs that do not embed it.*
F1279 Tlamatini/agent/static/agent/vendor/pdfjs/standard_fonts/LICENSE_FOXIT [27 lines] *Third-party license text: // Copyright 2014 PDFium Authors.*

F1280 `Tlamatini/agent/static/agent/vendor/pdfjs/standard_fonts/LICENSE_LIBERATION` [363 lines] *Third-party license text: LICENSE AGREEMENT AND LIMITED PRODUCT WARRANTY*

vendor/pdfjs/wasm — sheet 19

F1282 `Tlamatini/agent/static/agent/vendor/pdfjs/wasm/jbig2.wasm` [102KB] *PDF.js WebAssembly decoder module.*
F1283 `Tlamatini/agent/static/agent/vendor/pdfjs/wasm/jbig2_nowasm_fallback.js` [15 lines] *Vendored PDF.js WebAssembly decoders (JBIG2, OpenJPEG, QuickJS): jbig2_nowasm_fallback.js.*
F1284 `Tlamatini/agent/static/agent/vendor/pdfjs/wasm/LICENSE_JBIG2` [196 lines] *Third-party license text: // Copyright 2014 The PDFium Authors*
F1285 `Tlamatini/agent/static/agent/vendor/pdfjs/wasm/LICENSE_OPENJPEG` [39 lines] *Third-party license text: /*
F1286 `Tlamatini/agent/static/agent/vendor/pdfjs/wasm/LICENSE_PDFJS_JBIG2` [13 lines] *Third-party license text: Copyright 2026 Mozilla Foundation*
F1287 `Tlamatini/agent/static/agent/vendor/pdfjs/wasm/LICENSE_PDFJS_OPENJPEG` [22 lines] *Third-party license text: Copyright (c) 2024, Mozilla Foundation*
F1288 `Tlamatini/agent/static/agent/vendor/pdfjs/wasm/LICENSE_PDFJS_QCMS` [19 lines] *Third-party license text: MIT License*
F1289 `Tlamatini/agent/static/agent/vendor/pdfjs/wasm/LICENSE_QCMS` [21 lines] *Third-party license text: qcms*
F1290 `Tlamatini/agent/static/agent/vendor/pdfjs/wasm/openjpeg.wasm` [246KB] *PDF.js WebAssembly decoder module.*
F1291 `Tlamatini/agent/static/agent/vendor/pdfjs/wasm/openjpeg_nowasm_fallback.js` [17 lines] *Vendored PDF.js WebAssembly decoders (JBIG2, OpenJPEG, QuickJS): openjpeg_nowasm_fallback.js.*
F1292 `Tlamatini/agent/static/agent/vendor/pdfjs/wasm/qcms_bg.wasm` [94KB] *PDF.js WebAssembly decoder module.*
F1293 `Tlamatini/agent/static/agent/vendor/pdfjs/wasm/quickjs-eval.js` [16 lines] *Vendored PDF.js WebAssembly decoders (JBIG2, OpenJPEG, QuickJS): quickjs-eval.js.*
F1294 `Tlamatini/agent/static/agent/vendor/pdfjs/wasm/quickjs-eval.wasm` [458KB] *PDF.js WebAssembly decoder module.*

vendor/pdfjs/web — sheet 19

F1295 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/altText_add.svg` [3 lines] *PDF.js viewer icon: altText add.*
F1296 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/altText_disclaimer.svg` [3 lines] *PDF.js viewer icon: altText disclaimer.*
F1297 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/altText_done.svg` [3 lines] *PDF.js viewer icon: altText done.*
F1298 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/altText_spinner.svg` [30 lines] *PDF.js viewer icon: altText spinner.*
F1299 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/altText_warning.svg` [3 lines] *PDF.js viewer icon: altText warning.*
F1300 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/annotation-check.svg` [11 lines] *PDF.js viewer icon: annotation check.*
F1301 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/annotation-comment.svg` [16 lines] *PDF.js viewer icon: annotation comment.*
F1302 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/annotation-help.svg` [26 lines] *PDF.js viewer icon: annotation help.*
F1303 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/annotation-insert.svg` [10 lines] *PDF.js viewer icon: annotation insert.*
F1304 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/annotation-key.svg` [11 lines] *PDF.js viewer icon: annotation key.*
F1305 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/annotation-newparagraph.svg` [11 lines] *PDF.js viewer icon: annotation newparagraph.*
F1306 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/annotation-noicon.svg` [7 lines] *PDF.js viewer icon: annotation noicon.*
F1307 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/annotation-note.svg` [42 lines] *PDF.js viewer icon: annotation note.*
F1308 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/annotation-paperclip.svg` [6 lines] *PDF.js viewer icon: annotation paperclip.*
F1309 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/annotation-paragraph.svg` [16 lines] *PDF.js viewer icon: annotation paragraph.*
F1310 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/annotation-pushpin.svg` [7 lines] *PDF.js viewer icon: annotation pushpin.*
F1311 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/checkmark.svg` [5 lines] *PDF.js viewer icon: checkmark.*
F1312 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/comment-actionsButton.svg` [3 lines] *PDF.js viewer icon: comment actionsButton.*
F1313 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/comment-closeButton.svg` [3 lines] *PDF.js viewer icon: comment closeButton.*
F1314 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/comment-editButton.svg` [8 lines] *PDF.js viewer icon: comment editButton.*
F1315 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/comment-popup-editButton.svg` [5 lines] *PDF.js viewer icon: comment popup editButton.*
F1316 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/cursor-editorFreeHighlight.svg` [6 lines] *PDF.js viewer icon: cursor editorFreeHighlight.*
F1317 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/cursor-editorFreeText.svg` [3 lines] *PDF.js viewer icon: cursor editorFreeText.*
F1318 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/cursor-editorInk.svg` [4 lines] *PDF.js viewer icon: cursor editorInk.*
F1319 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/cursor-editorTextHighlight.svg` [8 lines] *PDF.js viewer icon: cursor editorTextHighlight.*
F1320 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/editor-toolbar-delete.svg` [5 lines] *PDF.js viewer icon: editor toolbar delete.*
F1321 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/editor-toolbar-edit.svg` [3 lines] *PDF.js viewer icon: editor toolbar edit.*
F1322 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/findbarButton-next.svg` [3 lines] *PDF.js viewer icon: findbarButton next.*
F1323 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/findbarButton-previous.svg` [3 lines] *PDF.js viewer icon: findbarButton previous.*
F1324 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/gv-toolbarButton-download.svg` [3 lines] *PDF.js viewer icon: gv toolbarButton download.*
F1325 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/loading-icon.gif` [2KB] *Image (GIF): loading-icon.*
F1326 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/loading.svg` [1 lines] *PDF.js viewer icon: loading.*
F1327 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/messageBar_closingButton.svg` [3 lines] *PDF.js viewer icon: messageBar closingButton.*
F1328 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/messageBar_info.svg` [3 lines] *PDF.js viewer icon: messageBar info.*
F1329 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/messageBar_warning.svg` [3 lines] *PDF.js viewer icon: messageBar warning.*
F1330 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/pages_closeButton.svg` [3 lines] *PDF.js viewer icon: pages closeButton.*
F1331 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/pages_selected.svg` [7 lines] *PDF.js viewer icon: pages selected.*
F1332 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/pages_viewArrow.svg` [3 lines] *PDF.js viewer icon: pages viewArrow.*
F1333 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/pages_viewButton.svg` [3 lines] *PDF.js viewer icon: pages viewButton.*
F1334 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/secondaryToolbarButton-documentProperties.svg` [3 lines] *PDF.js viewer icon: secondaryToolbarButton documentProperties.*
F1335 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/secondaryToolbarButton-firstPage.svg` [3 lines] *PDF.js viewer icon: secondaryToolbarButton firstPage.*
F1336 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/secondaryToolbarButton-handTool.svg` [3 lines] *PDF.js viewer icon: secondaryToolbarButton handTool.*
F1337 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/secondaryToolbarButton-lastPage.svg` [3 lines] *PDF.js viewer icon: secondaryToolbarButton lastPage.*
F1338 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/secondaryToolbarButton-rotateCcw.svg` [3 lines] *PDF.js viewer icon: secondaryToolbarButton rotateCcw.*
F1339 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/secondaryToolbarButton-rotateCw.svg` [3 lines] *PDF.js viewer icon: secondaryToolbarButton rotateCw.*
F1340 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/secondaryToolbarButton-scrollHorizontal.svg` [3 lines] *PDF.js viewer icon: secondaryToolbarButton scrollHorizontal.*
F1341 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/secondaryToolbarButton-scrollPage.svg` [3 lines] *PDF.js viewer icon: secondaryToolbarButton scrollPage.*
F1342 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/secondaryToolbarButton-scrollVertical.svg` [3 lines] *PDF.js viewer icon: secondaryToolbarButton scrollVertical.*
F1343 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/secondaryToolbarButton-scrollWrapped.svg` [3 lines] *PDF.js viewer icon: secondaryToolbarButton scrollWrapped.*
F1344 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/secondaryToolbarButton-selectTool.svg` [3 lines] *PDF.js viewer icon: secondaryToolbarButton selectTool.*
F1345 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/secondaryToolbarButton-spreadEven.svg` [3 lines] *PDF.js viewer icon: secondaryToolbarButton spreadEven.*
F1346 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/secondaryToolbarButton-spreadNone.svg` [3 lines] *PDF.js viewer icon: secondaryToolbarButton spreadNone.*
F1347 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/secondaryToolbarButton-spreadOdd.svg` [3 lines] *PDF.js viewer icon: secondaryToolbarButton spreadOdd.*
F1348 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/signature-properties-row-check.svg` [4 lines] *PDF.js viewer icon: signature properties row check.*
F1349 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-bookmark.svg` [3 lines] *PDF.js viewer icon: toolbarButton bookmark.*
F1350 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-currentOutlineItem.svg` [3 lines] *PDF.js viewer icon: toolbarButton currentOutlineItem.*
F1351 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-download.svg` [4 lines] *PDF.js viewer icon: toolbarButton download.*
F1352 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-editorFreeText.svg` [5 lines] *PDF.js viewer icon: toolbarButton editorFreeText.*
F1353 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-editorHighlight.svg` [6 lines] *PDF.js viewer icon: toolbarButton editorHighlight.*
F1354 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-editorInk.svg` [4 lines] *PDF.js viewer icon: toolbarButton editorInk.*
F1355 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-editorSignature.svg` [6 lines] *PDF.js viewer icon: toolbarButton editorSignature.*
F1356 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-editorStamp.svg` [8 lines] *PDF.js viewer icon: toolbarButton editorStamp.*
F1357 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-menuArrow.svg` [3 lines] *PDF.js viewer icon: toolbarButton menuArrow.*
F1358 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-menuArrowNova.svg` [3 lines] *PDF.js viewer icon: toolbarButton menuArrowNova.*
F1359 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-openFile.svg` [3 lines] *PDF.js viewer icon: toolbarButton openFile.*
F1360 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-pageDown.svg` [3 lines] *PDF.js viewer icon: toolbarButton pageDown.*
F1361 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-pageUp.svg` [3 lines] *PDF.js viewer icon: toolbarButton pageUp.*
F1362 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-presentationMode.svg` [3 lines] *PDF.js viewer icon: toolbarButton presentationMode.*
F1363 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-print.svg` [3 lines] *PDF.js viewer icon: toolbarButton print.*
F1364 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-search.svg` [3 lines] *PDF.js viewer icon: toolbarButton search.*
F1365 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-secondaryToolbarToggle.svg` [3 lines] *PDF.js viewer icon: toolbarButton secondaryToolbarToggle.*
F1366 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-signaturePropertiesError.svg` [7 lines] *PDF.js viewer icon: toolbarButton signaturePropertiesError.*
F1367 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-signaturePropertiesVerified.svg` [7 lines] *PDF.js viewer icon: toolbarButton signaturePropertiesVerified.*
F1368 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-viewAttachments.svg` [3 lines] *PDF.js viewer icon: toolbarButton viewAttachments.*
F1369 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-viewLayers.svg` [3 lines] *PDF.js viewer icon: toolbarButton viewLayers.*
F1370 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-viewOutline.svg` [3 lines] *PDF.js viewer icon: toolbarButton viewOutline.*
F1371 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-viewsManagerToggle.svg` [3 lines] *PDF.js viewer icon: toolbarButton viewsManagerToggle.*
F1372 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-viewThumbnail.svg` [3 lines] *PDF.js viewer icon: toolbarButton viewThumbnail.*
F1373 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-zoomIn.svg` [3 lines] *PDF.js viewer icon: toolbarButton zoomIn.*
F1374 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/toolbarButton-zoomOut.svg` [3 lines] *PDF.js viewer icon: toolbarButton zoomOut.*
F1375 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/treeitem-collapsed.svg` [1 lines] *PDF.js viewer icon: treeitem collapsed.*
F1376 `Tlamatini/agent/static/agent/vendor/pdfjs/web/images/treeitem-expanded.svg` [1 lines] *PDF.js viewer icon: treeitem expanded.*
F1377 `Tlamatini/agent/static/agent/vendor/pdfjs/web/pdf_viewer.css` [6.4k lines] *Vendored PDF.js viewer and its toolbar icons: pdf_viewer.css.*
F1378 `Tlamatini/agent/static/agent/vendor/pdfjs/web/pdf_viewer.mjs` [9.8k lines] *Vendored PDF.js viewer and its toolbar icons: pdf_viewer.mjs.*

Atlas VI — Agent pool: every workflow agent template

agents/ · shared helpers — sheet 20

F0615 `Tlamatini/agent/agents/model_settings.py` [166 lines] *Shared model registry for the Models dialog and standalone workflow agents.*
F0732 `Tlamatini/agent/agents/visual_errors.py` [33 lines] *Report a failed visual-analysis attempt without controlling recovery or retries.*

Cleaner — sheet 20

F0516 `Tlamatini/agent/agents/cleaner/__init__.py` [9 lines] *Package marker for cleaner/.*
F0517 `Tlamatini/agent/agents/cleaner/cleaner.py` [335 lines] *Cleaner Agent - Cleans up logs/pids of target agents and triggers restarts This agent is designed to run AFTER an Ender agent.*
F0518 `Tlamatini/agent/agents/cleaner/config.yaml` [10 lines] *Default config.yaml of the Cleaner agent: agents_to_clean, cleaned_agents, output_agents.*

Croner — sheet 20

F0523 `Tlamatini/agent/agents/croner/config.yaml` [7 lines] *Default config.yaml of the Croner agent: trigger_time, poll_interval.*
F0524 `Tlamatini/agent/agents/croner/croner.py` [414 lines] *Croner Agent - Deterministic Time/Pattern Trigger Agent Triggers target agents at a specific scheduled time, optionally requiring a specific pattern. . .*

Ender — sheet 20

F0537 `Tlamatini/agent/agents/ender/config.yaml` [29 lines] *Default config.yaml of the Ender agent: output_agents.*
F0538 `Tlamatini/agent/agents/ender/ender.py` [660 lines] *Ender Agent - No LLM, deterministic agent terminator This agent terminates all agents listed in target_agents when triggered.*

Sleeper — sheet 20

F0687 `Tlamatini/agent/agents/sleeper/config.yaml` [3 lines] *Default config.yaml of the Sleeper agent: duration_ms.*
F0688 `Tlamatini/agent/agents/sleeper/sleeper.py` [393 lines] *Sleeper Agent - Deterministic agent to wait and then trigger downstream Action: Triggered by upstream -> Wait -> Trigger downstream*

Starter — sheet 20

F0693 `Tlamatini/agent/agents/starter/config.yaml` [25 lines] *Default config.yaml of the Starter agent: exit_after_start.*
F0694 `Tlamatini/agent/agents/starter/starter.py` [495 lines] *Starter Agent - No LLM, deterministic agent starter This agent starts all agents connected to its output when executed.*

Stopper — sheet 20

F0697 `Tlamatini/agent/agents/stopper/config.yaml` [53 lines] *Default config.yaml of the Stopper agent: patterns, output_agents, poll_interval.*
F0698 `Tlamatini/agent/agents/stopper/stopper.py` [451 lines] *Stopper Agent - Single-threaded pattern-based agent terminator This agent monitors source agent log files for configurable patterns and terminates. . .*

Asker — sheet 20

F0505 `Tlamatini/agent/agents/asker/asker.py` [469 lines] *Asker Agent - Deterministic agent that asks the user to choose a path When started, writes ASKER_CHOICE_NEEDED to log.*
F0506 `Tlamatini/agent/agents/asker/config.yaml` [34 lines] *Default config.yaml of the Asker agent: target_agents_a, target_agents_b, legend_path_a, legend_path_b.*

Counter — sheet 20

F0519 `Tlamatini/agent/agents/counter/config.yaml` [44 lines] *Default config.yaml of the Counter agent: initial_value, threshold_value, target_agents_l, target_agents_g.*
F0520 `Tlamatini/agent/agents/counter/counter.py` [450 lines] *Counter Agent - Deterministic agent that maintains a persistent counter and routes to Path L (less than) or Path G (greater than or equal) based on. . .*

Forker — sheet 20

F0575 `Tlamatini/agent/agents/forker/config.yaml` [42 lines] *Default config.yaml of the Forker agent: pattern_a, pattern_b, target_agents_a, target_agents_b, poll_interval.*
F0576 `Tlamatini/agent/agents/forker/forker.py` [656 lines] *Forker Agent - Deterministic agent that monitors source agent logs and automatically routes to Path A or Path B based on pattern detection.*

Raiser — sheet 20

F0675 `Tlamatini/agent/agents/raiser/config.yaml` [45 lines] *Default config.yaml of the Raiser agent: pattern, poll_interval.*
F0676 `Tlamatini/agent/agents/raiser/raiser.py` [652 lines] *Raiser Agent - No LLM, deterministic event-driven agent launcher This agent monitors log files of source agents for a configurable pattern string. . .*

AND — sheet 20

F0494 `Tlamatini/agent/agents/and/and.py` [542 lines] *AND Agent - Deterministic AND gate logic Monitors two source log files for specific patterns.*
F0495 `Tlamatini/agent/agents/and/config.yaml` [17 lines] *Default config.yaml of the AND agent: source_agent_1, pattern_1, source_agent_2, pattern_2, poll_interval.*

Barrier — sheet 20

F0509 `Tlamatini/agent/agents/barrier/barrier.py` [507 lines] *Barrier Agent - OS-process-level synchronization barrier for flow control Each source agent STARTS a separate barrier process ("input sub-process").*
F0510 `Tlamatini/agent/agents/barrier/config.yaml` [8 lines] *Default config.yaml of the Barrier agent: .*
F0511 `Tlamatini/agent/agents/barrier/test_barrier.py` [278 lines] *Test script for the Barrier agent.*

OR — sheet 20

F0633 `Tlamatini/agent/agents/or/config.yaml` [16 lines] *Default config.yaml of the OR agent: source_agent_1, pattern_1, source_agent_2, pattern_2, poll_interval.*
F0634 `Tlamatini/agent/agents/or/or.py` [517 lines] *OR Agent - Deterministic OR gate logic Monitors two source log files for specific patterns.*

Analyzer — sheet 20

F0492 `Tlamatini/agent/agents/analyzer/analyzer.py` [540 lines] *Analyzer Agent - deterministic static-analysis / security scanner Action: Triggered by upstream -> run every installed scanner over target_path ->. . .*
F0493 `Tlamatini/agent/agents/analyzer/config.yaml` [33 lines] *Default config.yaml of the Analyzer agent: target_path, tools, min_severity, max_report_chars.*

Apirer — sheet 20

F0496 `Tlamatini/agent/agents/apirer/apirer.py` [510 lines] *Apirer Agent - HTTP/REST API Agent Action: Triggered by upstream -> Make HTTP request -> Log response -> Trigger downstream (always)*
F0497 `Tlamatini/agent/agents/apirer/config.yaml` [10 lines] *Default config.yaml of the Apirer agent: url, method, headers, body, expected_status, timeout.*

Arduiner — sheet 20

F0498 `Tlamatini/agent/agents/arduiner/arduiner.py` [1.5k lines] *Arduiner Agent - Arduino CLI (arduino-cli) bridge (firmware scaffold/build/upload/monitor) Action: Triggered by upstream -> resolve the arduino-cli. . .*
F0499 `Tlamatini/agent/agents/arduiner/ArduinoTemplateProject/ArduinoTemplateProject.ino` [40 lines] *ArduinoTemplateProject.ino — Tlamatini Arduiner template sketch.*
F0500 `Tlamatini/agent/agents/arduiner/ArduinoTemplateProject/README.md` [53 lines] *ArduinoTemplateProject: The bundled scaffold the Arduiner agent (Tlamatini) copies when you run action='create_project'.*
F0501 `Tlamatini/agent/agents/arduiner/ArduinoTemplateProject/sketch.yaml` [23 lines] *sketch.yaml — Arduino CLI sketch project file (the Arduino-native peer of PlatformIO's platformio.ini).*
F0502 `Tlamatini/agent/agents/arduiner/ArduinoTemplateProject/src/Heartbeat.cpp` [30 lines] *Firmware source of the template project.*
F0503 `Tlamatini/agent/agents/arduiner/ArduinoTemplateProject/src/Heartbeat.h` [35 lines] *Heartbeat.h — a tiny, board-agnostic non-blocking LED blinker.*
F0504 `Tlamatini/agent/agents/arduiner/config.yaml` [125 lines] *Default config.yaml of the Arduiner agent: action, arduino_cli_executable, auto_bootstrap, arduino_cli_install_dir, auto_update, preflight, fqbn +16 more.*

AudioPlayer — sheet 20

F0507 `Tlamatini/agent/agents/audioplayer/audioplayer.py` [839 lines] *AudioPlayer Agent - Plays an audio FILE through a system OUTPUT device.*
F0508 `Tlamatini/agent/agents/audioplayer/config.yaml` [74 lines] *Default config.yaml of the AudioPlayer agent: audio_file, device_index, device_name, volume_percent, time_played, sample_rate.*

Blenderer — sheet 20

F0512 `Tlamatini/agent/agents/blenderer/blenderer.py` [742 lines] *Blenderer Agent — drives Blender via the OFFICIAL Blender MCP add-on's TCP socket protocol (localhost:9876 by default. . .*
F0513 `Tlamatini/agent/agents/blenderer/config.yaml` [63 lines] *Default config.yaml of the Blenderer agent: host, port, command, strict_json, params, connect_timeout, read_timeout.*

Camcorder — sheet 20

F0514 `Tlamatini/agent/agents/camcorder/camcorder.py` [674 lines] *Camcorder Agent - Camera capture agent (photo or video) via OpenCV Action: Triggered by upstream -> Open a system camera -> Take ONE photo (default). . .*
F0515 `Tlamatini/agent/agents/camcorder/config.yaml` [44 lines] *Default config.yaml of the Camcorder agent: camera_index, capture_mode, video_duration_seconds, video_fps, resolution_width, resolution_height, warmup_seconds +1 more.*

Crawler — sheet 20

F0521 `Tlamatini/agent/agents/crawler/config.yaml` [39 lines] *Default config.yaml of the Crawler agent: url, urls, system_prompt, content_mode, include_headers, crawl_type, include_seed +9 more.*
F0522 `Tlamatini/agent/agents/crawler/crawler.py` [2.2k lines] *Crawler Agent - Web page crawler with RAW content capture and LLM analysis Action: Triggered by upstream -> Fetch URL -> Capture RAW HTTP response. . .*

De-Compresser — sheet 20

F0525 `Tlamatini/agent/agents/de_compresser/config.yaml` [26 lines] *Default config.yaml of the De-Compresser agent: input, output, passwordless.*
F0526 `Tlamatini/agent/agents/de_compresser/de_compresser.py` [848 lines] *De-Compresser Agent - Deterministic compression / decompression*

Deleter — sheet 20

F0527 `Tlamatini/agent/agents/deleter/config.yaml` [29 lines] *Default config.yaml of the Deleter agent: trigger_mode, target_path, files_to_delete, allow_directory_delete, recursive, filetype_exclusions, trigger_event_string +1 more.*
F0528 `Tlamatini/agent/agents/deleter/deleter.py` [764 lines] *Deleter Agent - Deterministic agent to delete files Triggers: Immediate or Event-based (Source Log) Action: Delete files based on patterns (supports. . .*

Discoverer — sheet 20

F0529 `Tlamatini/agent/agents/discoverer/config.yaml` [114 lines] *Default config.yaml of the Discoverer agent: tool, target, targets_file, json_output, output_dir, rate_limit, concurrency +34 more.*
F0530 `Tlamatini/agent/agents/discoverer/discoverer.py` [1.3k lines] *Discoverer Agent - ProjectDiscovery suite bridge (subfinder/httpx/naabu/katana/nuclei/cvemap) Action: Triggered by upstream -> resolve the requested. . .*

Dockerer — sheet 20

F0531 `Tlamatini/agent/agents/dockerer/config.yaml` [3 lines] *Default config.yaml of the Dockerer agent: command.*
F0532 `Tlamatini/agent/agents/dockerer/dockerer.py` [519 lines] *Dockerer Agent - Docker container management agent Action: Triggered by upstream -> Execute docker command -> Trigger downstream (on success or error)*

Editor — sheet 20

F0533 `Tlamatini/agent/agents/editor/config.yaml` [16 lines] *Default config.yaml of the Editor agent: file_path, old_string, new_string, old_string_b64, new_string_b64, replace_all.*
F0534 `Tlamatini/agent/agents/editor/editor.py` [409 lines] *Editor Agent - Surgical in-place find-and-replace on a single text file Action: Triggered by upstream -> edit file -> emit INI_SECTION_EDITOR ->. . .*

ESP32er — sheet 20

F0539 `Tlamatini/agent/agents/esp32er/config.yaml` [109 lines] *Default config.yaml of the ESP32er agent: action, pio_executable, auto_bootstrap, pio_install_method, pio_core_dir, auto_update, pip_install +13 more.*
F0540 `Tlamatini/agent/agents/esp32er/esp32er.py` [1.4k lines] *ESP32er Agent - PlatformIO Core (pio CLI) bridge (firmware scaffold/build/upload/monitor) Action: Triggered by upstream -> resolve the pio. . .*
F0541 `Tlamatini/agent/agents/esp32er/ESP32TemplateProject/include/README.md` [16 lines] *include/ — project header files: Put your own .h / .hpp files here.*
F0542 `Tlamatini/agent/agents/esp32er/ESP32TemplateProject/lib/README.md` [35 lines] *lib/ — private project libraries: Each subdirectory here is compiled as its own static library and linked in automatically.*
F0543 `Tlamatini/agent/agents/esp32er/ESP32TemplateProject/platformio.ini` [40 lines] *; . . .*
F0544 `Tlamatini/agent/agents/esp32er/ESP32TemplateProject/README.md` [114 lines] *ESP32TemplateProject: A known-good ESP32 firmware baseline: it blinks the onboard LED and prints LED ON / LED OFF over serial at 115200 baud.*
F0545 `Tlamatini/agent/agents/esp32er/ESP32TemplateProject/src/main.cpp` [57 lines] *ESP32TemplateProject — the whole firmware.*
F0546 `Tlamatini/agent/agents/esp32er/ESP32TemplateProject/test/README.md` [32 lines] *test/ — unit tests: PlatformIO's test runner (Unity) picks up tests from this directory.*

ESPHomer — sheet 20

F0547 `Tlamatini/agent/agents/esphomer/config.yaml` [104 lines] *Default config.yaml of the ESPHomer agent: action, esphome_executable, auto_bootstrap, pip_install, auto_update, preflight, config_path +12 more.*
F0548 `Tlamatini/agent/agents/esphomer/esphomer.py` [1.4k lines] *ESPHomer Agent - ESPHome (esphome CLI) bridge (device YAML author/validate/compile/upload/observe) Action: Triggered by upstream -> resolve the. . .*
F0549 `Tlamatini/agent/agents/esphomer/ESPHomeTemplateProject/.gitignore` [8 lines] *Your real credentials — NEVER commit these.*
F0550 `Tlamatini/agent/agents/esphomer/ESPHomeTemplateProject/common/base.yaml` [32 lines] *Shared building blocks every device in this project pulls in with*
F0551 `Tlamatini/agent/agents/esphomer/ESPHomeTemplateProject/README.md` [140 lines] *ESPHomeTemplateProject: A known-good ESPHome smart-home baseline — two working devices, shared configuration, and credentials kept out of git.*
F0552 `Tlamatini/agent/agents/esphomer/ESPHomeTemplateProject/secrets.yaml.example` [15 lines] *. . .*
F0553 `Tlamatini/agent/agents/esphomer/ESPHomeTemplateProject/tlamatini-light.yaml` [47 lines] *Tlamatini ESPHome sample device — a phone-controlled light.*

F0554 `Tlamatini/agent/agents/esphomer/ESPHomeTemplateProject/tlamatini-sensor.yaml` [70 lines] *Tlamatini ESPHome sample device #2 — a sensor node, the "grown-up" shape.*

Executer — sheet 20

F0555 `Tlamatini/agent/agents/executer/config.yaml` [40 lines] *Default config.yaml of the Executer agent: script, non_blocking, command_timeout_seconds, execute_forked_window.*
F0556 `Tlamatini/agent/agents/executer/executer.py` [1.1k lines] *Executer Agent - Deterministic agent to execute a command Logs EXECUTION SUCCESS or EXECUTION FAILED based on result Always triggers downstream...*

File-Creator — sheet 20

F0557 `Tlamatini/agent/agents/file_creator/config.yaml` [22 lines] *Default config.yaml of the File-Creator agent: file_path, content, content_b64, append.*
F0558 `Tlamatini/agent/agents/file_creator/file_creator.py` [463 lines] *File-Creator Agent - Creates a file with specified content, triggers downstream agents, then stops Action: Triggered by upstream -> Write file ->...*

File-Extractor — sheet 20

F0559 `Tlamatini/agent/agents/file_extractor/config.yaml` [10 lines] *Default config.yaml of the File-Extractor agent: path_filenames, recursive, filetype_exclusions, line_numbers, offset, limit.*
F0560 `Tlamatini/agent/agents/file_extractor/file_extractor.py` [867 lines] *File-Extractor Agent - Extracts text content from files, triggers downstream agents, then stops Action: Triggered by upstream -> Read/extract...*

File-Interpreter — sheet 20

F0561 `Tlamatini/agent/agents/file_interpreter/config.yaml` [46 lines] *Default config.yaml of the File-Interpreter agent: path_filenames, recursive, filetype_exclusions, reading_type, llm.*
F0562 `Tlamatini/agent/agents/file_interpreter/file_interpreter.py` [1.0k lines] *File-Interpreter Agent - Document parsing and text/image extraction Hybrid deterministic/non-deterministic agent Action: Read files -> Extract text...*

Gitter — sheet 20

F0577 `Tlamatini/agent/agents/gitter/config.yaml` [27 lines] *Default config.yaml of the Gitter agent: repo_path, command, branch, commit_message, remote, custom_command.*
F0578 `Tlamatini/agent/agents/gitter/gitter.py` [533 lines] *Gitter Agent - Git operations agent Action: Triggered by upstream -> Execute git command -> Log output -> Trigger downstream*

Globber — sheet 20

F0579 `Tlamatini/agent/agents/globber/config.yaml` [14 lines] *Default config.yaml of the Globber agent: pattern, path, sort_by, max_results.*
F0580 `Tlamatini/agent/agents/globber/globber.py` [702 lines] *Globber Agent - Read-only filename pattern search (find files by glob) Action: Triggered by upstream -> glob -> emit INI_SECTION_GLOBBER -> Trigger...*

Googler — sheet 20

F0581 `Tlamatini/agent/agents/googler/config.yaml` [208 lines] *Default config.yaml of the Googler agent: query, preset, filetypes, filetype, sites, site, exclude_sites +36 more.*
F0582 `Tlamatini/agent/agents/googler/googler.py` [2.0k lines] *Googler Agent - resilient indexed-web search agent with content extraction Action: Triggered by upstream -> Search the engine chain -> Fetch top N...*
F0583 `Tlamatini/agent/agents/googler/googler_engines.py` [1.4k lines] *Googler resilience core: the part of Googler that must never hang and never lie.*

Grepper — sheet 20

F0584 `Tlamatini/agent/agents/grepper/config.yaml` [31 lines] *Default config.yaml of the Grepper agent: pattern, path, glob, case_insensitive, output_mode, max_results, start_line +2 more.*
F0585 `Tlamatini/agent/agents/grepper/grepper.py` [894 lines] *Grepper Agent - Read-only regex content search across a file or directory tree Action: Triggered by upstream -> search -> emit INI_SECTION_GREPPER...*

Image-Interpreter — sheet 20

F0586 `Tlamatini/agent/agents/image_interpreter/config.yaml` [156 lines] *Default config.yaml of the Image-Interpreter agent: images_pathfilenames, recursive, filetype_exclusions, interpreter_model_1, interpreter_model_2, merging_model, prompt_interpreter_model_1 +4 more.*
F0587 `Tlamatini/agent/agents/image_interpreter/image_interpreter.py` [940 lines] *Image-Interpreter Agent - TRIPLE-MODEL image analysis via LLM vision models Non-deterministic agent (uses LLMs) Action: Resolve image paths ->...*

Instant Messaging Doctor — sheet 20

F0588 `Tlamatini/agent/agents/instant_messaging_doctor/config.yaml` [52 lines] *Default config.yaml of the Instant Messaging Doctor agent: mode, platform, contact_name, message, telegram, whatsapp, template +10 more.*
F0589 `Tlamatini/agent/agents/instant_messaging_doctor/instant_messaging_doctor.py` [706 lines] *Instant Messaging Doctor Agent - Telegrammer/Whatsapper diagnostics*

J-Decompiler — sheet 20

F0590 `Tlamatini/agent/agents/j_decompiler/config.yaml` [11 lines] *Default config.yaml of the J-Decompiler agent: directory, recursive.*
F0591 `Tlamatini/agent/agents/j_decompiler/j_decompiler.py` [514 lines] *J-Decompiler Agent - Decompile .class, .jar, .war, .ear files using jd-cli Action: Triggered by upstream -> Decompile Java files -> Trigger downstream*

Jenkinser — sheet 20

F0592 `Tlamatini/agent/agents/jenkinser/config.yaml` [21 lines] *Default config.yaml of the Jenkinser agent: jenkins_url, job_name, user, api_token, parameters, use_parameters.*
F0593 `Tlamatini/agent/agents/jenkinser/jenkinser.py` [491 lines] *Jenkinser Agent - CI/CD Pipeline Trigger Action: Triggered by upstream -> Trigger Jenkins build -> Log result -> Trigger downstream (always)*

Kalier — sheet 20

F0594 `Tlamatini/agent/agents/kalier/config.yaml` [104 lines] *Default config.yaml of the Kalier agent: action, server_url, timeout, OOB_shift_reaper, hang_detect_idle_seconds, target, url +16 more.*
F0595 `Tlamatini/agent/agents/kalier/kalier.py` [703 lines] *Kalier Agent - MCP-Kali-Server bridge (AI-assisted penetration testing) Action: Triggered by upstream -> POST to the MCP-Kali-Server Flask API...*

Keyboarder — sheet 20

F0596 `Tlamatini/agent/agents/keyboarder/config.yaml` [26 lines] *Default config.yaml of the Keyboarder agent: stride_delay, typing_interval_ms, window_title, window_handle, window_match_index, input_mode, text +1 more.*
F0597 `Tlamatini/agent/agents/keyboarder/keyboarder.py` [570 lines] *Keyboarder Agent - Issues keyboard sequences simulating the user pressing keys*

F0598 `Tlamatini/agent/agents/keyboarder/keyboarder_input.py` [188 lines] *Checked Windows keyboard input without changing the clipboard.*

Kuberneter — sheet 20

F0599 `Tlamatini/agent/agents/kuberneter/config.yaml` [17 lines] *Default config.yaml of the Kuberneter agent: command, namespace, extra_args, custom_command.*
F0600 `Tlamatini/agent/agents/kuberneter/kuberneter.py` [411 lines] *Kuberneter Agent - Execute kubectl commands and trigger downstream Action: Triggered by upstream -> Execute kubectl command -> Trigger downstream...*

LaTeXer — sheet 20

F0607 `Tlamatini/agent/agents/latexer/config.yaml` [221 lines] *Default config.yaml of the LaTeXer agent: action, tex_path, project_dir, main_file, input_text, input_text_b64, recursive +55 more.*
F0608 `Tlamatini/agent/agents/latexer/latexer.py` [5.3k lines] *LaTeXer Agent - Tlamatini's LaTeX TYPESETTING agent (author + compile .tex -> PDF).*
F0609 `Tlamatini/agent/agents/latexer/latexer_artwork.py` [273 lines] *Original, deterministic TikZ artwork.*
F0610 `Tlamatini/agent/agents/latexer/latexer_design.py` [267 lines] *Portable styled LaTeX scaffolds, with separate document and Beamer recipes.*
F0611 `Tlamatini/agent/agents/latexer/latexer_styles.py` [154 lines] *LaTeXer's portable design catalogue.*
F0612 `Tlamatini/agent/agents/latexer/STYLES.md` [242 lines] *LaTeXer: the signature collection: 30 explicit visual identities, independent of the eight document templates.*

MCP Doctor — sheet 20

F0613 `Tlamatini/agent/agents/mcp_doctor/config.yaml` [10 lines] *Default config.yaml of the MCP Doctor agent: server_key, catalog_path, source_url, mode, include_catalog.*
F0614 `Tlamatini/agent/agents/mcp_doctor/mcp_doctor.py` [694 lines] *MCP Doctor Agent - External MCP catalog diagnostics and setup guidance*

Mongoxer — sheet 20

F0616 `Tlamatini/agent/agents/mongoxer/config.yaml` [18 lines] *Default config.yaml of the Mongoxer agent: mongo_connection, script.*
F0617 `Tlamatini/agent/agents/mongoxer/mongoxer.py` [196 lines] *conhost.exe orphan guard ——————— When Tlamatini's runtime launches us with DETACHED_PROCESS we have no console...*

Mouser — sheet 20

F0622 `Tlamatini/agent/agents/mouser/config.yaml` [88 lines] *Default config.yaml of the Mouser agent: movement_type, coordinate_space, capture_left, capture_top, capture_width, capture_height, image_width +16 more.*
F0623 `Tlamatini/agent/agents/mouser/mouser.py` [783 lines] *Mouser Agent - Mouse pointer movement agent Action: Triggered by upstream -> Move mouse (random or localized) -> Trigger downstream*
F0624 `Tlamatini/agent/agents/mouser/mouser_coordinates.py` [177 lines] *Physical desktop coordinates for Mouser (also copied to isolated runs).*

Mover — sheet 20

F0625 `Tlamatini/agent/agents/mover/config.yaml` [14 lines] *Default config.yaml of the Mover agent: trigger_mode, operation, source_files, destination_folder, recursive, filetype_exclusions, trigger_event_string +1 more.*
F0626 `Tlamatini/agent/agents/mover/mover.py` [870 lines] *Mover Agent - Deterministic agent to copy/move files Triggers: Immediate or Event-based (Source Log) Action: Move or Copy files based on patterns*

NetSpeed-Calculator — sheet 20

F0627 `Tlamatini/agent/agents/netspeed_calculator/config.yaml` [83 lines] *Default config.yaml of the NetSpeed-Calculator agent: action, providers, min_successful_providers, parallel_streams, test_duration_seconds, warmup_seconds, sample_interval_seconds +14 more.*
F0628 `Tlamatini/agent/agents/netspeed_calculator/netspeed_calculator.py` [1.9k lines] *NetSpeed-Calculator Agent — formal, multi-provider Internet throughput measurement.*

Nmapper — sheet 20

F0629 `Tlamatini/agent/agents/nmapper/config.yaml` [88 lines] *Default config.yaml of the Nmapper agent: action, target, targets_file, ports, top_ports, timing, min_rate +17 more.*
F0630 `Tlamatini/agent/agents/nmapper/nmapper.py` [1.2k lines] *Nmapper Agent - LOCAL, USE-ONLY nmap bridge for pentesters / CTF recon.*

PDFer — sheet 20

F0637 `Tlamatini/agent/agents/pdfer/config.yaml` [217 lines] *Default config.yaml of the PDFer agent: mode, input_text, input_file, images, input_pdfs, title, subtitle +45 more.*
F0638 `Tlamatini/agent/agents/pdfer/pdfer.py` [1.9k lines] *PDFer Agent - Tlamatini's DOCUMENT COMPOSER (the WRITE side of the document family).*
F0639 `Tlamatini/agent/agents/pdfer/pdfer_artwork.py` [261 lines] *Resolution-independent signature artwork, clipped to a text-free cover zone.*
F0640 `Tlamatini/agent/agents/pdfer/pdfer_atelier.py` [1.0k lines] *PDFer atelier — the ReportLab Platypus renderer that replaced xhtml2pdf.*
F0641 `Tlamatini/agent/agents/pdfer/pdfer_audit.py` [494 lines] *PDFer audit — checking the FILE ON DISK, not the renderer's opinion of it.*
F0642 `Tlamatini/agent/agents/pdfer/pdfer_color.py` [896 lines] *PDFer colour science — the pigment bench of the document composer.*
F0643 `Tlamatini/agent/agents/pdfer/pdfer_consult.py` [378 lines] *PDFer design consultation — asking Tlamatini's own model for an art direction.*
F0644 `Tlamatini/agent/agents/pdfer/pdfer_docmodel.py` [777 lines] *PDFer document model — content parsed into blocks a renderer can lay out.*
F0645 `Tlamatini/agent/agents/pdfer/pdfer_nuance.py` [1.2k lines] *PDFer nuance engine — reading the content BEFORE choosing how to dress it.*
F0646 `Tlamatini/agent/agents/pdfer/pdfer_ornament.py` [702 lines] *PDFer ornament — generated artwork, and the judgement about when to use it.*
F0647 `Tlamatini/agent/agents/pdfer/pdfer_styles.py` [104 lines] *Selectable visual identities, independent of the document's semantic nuance.*
F0648 `Tlamatini/agent/agents/pdfer/pdfer_tables.py` [664 lines] *PDFer table layout — the overlap fix, and why the old one could not work.*
F0649 `Tlamatini/agent/agents/pdfer/pdfer_theme.py` [1.2k lines] *PDFer themes — turning a nuance into a complete, legible design system.*
F0650 `Tlamatini/agent/agents/pdfer/pdfer_typography.py` [709 lines] *PDFer typography — the type case of the document composer.*
F0651 `Tlamatini/agent/agents/pdfer/STYLES.md` [239 lines] *PDFer: the signature collection: PDFer now offers 24 selectable visual identities alongside its 20 existing document themes.*

Playwrighter — sheet 20

F0652 `Tlamatini/agent/agents/playwrighter/config.yaml` [80 lines] *Default config.yaml of the Playwrighter agent: start_url, browser, headless, timeout_ms, nav_wait_until, user_agent, viewport_width +9 more.*
F0653 `Tlamatini/agent/agents/playwrighter/playwrighter.py` [1.0k lines] *Playwrighter Agent — scripted, interactive browser automation via Playwright.*

PPTXer — sheet 20

F0654 `Tlamatini/agent/agents/pptxer/config.yaml` [179 lines] *Default config.yaml of the PPTXer agent: input_text, input_file, input_format, title, subtitle, author, kicker +47 more.*

F0655 `Tlamatini/agent/agents/pptxer/pptxer.py` [1.2k lines] *PPTXer Agent — Tlamatini's PRESENTATION COMPOSER.*
F0656 `Tlamatini/agent/agents/pptxer/pptxer_audit.py` [645 lines] *pptxer_audit.py — DID THE DECK ACTUALLY COME OUT RIGHT?*
F0657 `Tlamatini/agent/agents/pptxer/pptxer_build.py` [1.5k lines] *pptxer_build.py — THE ASSEMBLY ENGINE.*
F0658 `Tlamatini/agent/agents/pptxer/pptxer_color.py` [736 lines] *pptxer_color.py — PPTXer's COLOUR SCIENCE.*
F0659 `Tlamatini/agent/agents/pptxer/pptxer_diagram.py` [433 lines] *pptxer_diagram.py — DIAGRAMS AS NATIVE, EDITABLE POWERPOINT SHAPES.*
F0660 `Tlamatini/agent/agents/pptxer/pptxer_docmodel.py` [731 lines] *pptxer_docmodel.py — CONTENT → DECK MODEL.*
F0661 `Tlamatini/agent/agents/pptxer/pptxer_draw.py` [866 lines] *pptxer_draw.py — GENERATED ARTWORK.*
F0662 `Tlamatini/agent/agents/pptxer/pptxer_fonts.py` [1.0k lines] *pptxer_fonts.py — PPTXer's TYPEFACE ENGINE.*
F0663 `Tlamatini/agent/agents/pptxer/pptxer_layout.py` [853 lines] *pptxer_layout.py — THE GEOMETRY SOLVER.*
F0664 `Tlamatini/agent/agents/pptxer/pptxer_media.py` [693 lines] *pptxer_media.py — MEDIA ACQUISITION AND PREPARATION.*
F0665 `Tlamatini/agent/agents/pptxer/pptxer_nuance.py` [887 lines] *pptxer_nuance.py — WHAT KIND OF DECK IS THIS?*
F0666 `Tlamatini/agent/agents/pptxer/pptxer_render.py` [944 lines] *pptxer_render.py — PPTXer LOOKS AT ITS OWN SLIDES.*
F0667 `Tlamatini/agent/agents/pptxer/pptxer_theme.py` [621 lines] *pptxer_theme.py — the DESIGN SYSTEM: nuance + seed colour → a complete, validated, machine-usable specification for every slide in the deck.*
F0668 `Tlamatini/agent/agents/pptxer/STYLES.md` [100 lines] *PPTXer presentation styles: PPTXer provides 36 named treatments: the original 24 content treatments and 12 additional visual styles.*

Prompter — sheet 20

F0669 `Tlamatini/agent/agents/prompter/config.yaml` [30 lines] *Default config.yaml of the Prompter agent: prompt, llm.*
F0670 `Tlamatini/agent/agents/prompter/prompter.py` [443 lines] *Prompter Agent - Sends a configured prompt to an Ollama LLM and logs the response Action: Triggered by upstream -> Read prompt from config -> Query...*

PSer — sheet 20

F0671 `Tlamatini/agent/agents/pser/config.yaml` [32 lines] *Default config.yaml of the PSer agent: likely_process_name, llm.*
F0672 `Tlamatini/agent/agents/pser/pser.py` [572 lines] *PSer Agent - Process finder agent with LLM-powered fuzzy matching Action: Triggered by upstream -> List processes (no admin) -> Ask LLM for best...*

Pythonxer — sheet 20

F0673 `Tlamatini/agent/agents/pythonxer/config.yaml` [46 lines] *Default config.yaml of the Pythonxer agent: script, execute_forked_window, ruff_blocking.*
F0674 `Tlamatini/agent/agents/pythonxer/pythonxer.py` [829 lines] *Pythonxer Agent - Deterministic agent to execute a Python script Validates with Ruff, executes the script, and triggers downstream agents ONLY if...*

Recorder — sheet 20

F0679 `Tlamatini/agent/agents/recorder/config.yaml` [57 lines] *Default config.yaml of the Recorder agent: device_index, device_name, record_seconds, sample_rate, channels, input_gain_percent, output_dir.*
F0680 `Tlamatini/agent/agents/recorder/recorder.py` [984 lines] *Recorder Agent - Microphone / audio-input capture agent via sounddevice Action: Triggered by upstream -> Open a system audio INPUT device (mic) ->...*

Reviewer — sheet 20

F0681 `Tlamatini/agent/agents/reviewer/config.yaml` [38 lines] *Default config.yaml of the Reviewer agent: repo_path, diff_ref, focus, max_diff_chars, llm.*
F0682 `Tlamatini/agent/agents/reviewer/reviewer.py` [588 lines] *Reviewer Agent - LLM-powered code review of a git diff Action: Triggered by upstream -> resolve git diff -> ask Ollama LLM to review it -> parse a...*

SCPer — sheet 20

F0683 `Tlamatini/agent/agents/scper/config.yaml` [27 lines] *Default config.yaml of the SCPer agent: user, ip, file, direction.*
F0684 `Tlamatini/agent/agents/scper/scper.py` [502 lines] *SCPer Agent - Deterministic agent to send/receive files via SCP Connects to a remote host and transfers a file.*

Shoter — sheet 20

F0685 `Tlamatini/agent/agents/shoter/config.yaml` [21 lines] *Default config.yaml of the Shoter agent: output_dir, all_screens, filename.*
F0686 `Tlamatini/agent/agents/shoter/shoter.py` [585 lines] *Shoter Agent - Screenshot capture agent Action: Triggered by upstream -> Capture screenshot -> Save to numbered dir -> Trigger downstream*

SQLer — sheet 20

F0689 `Tlamatini/agent/agents/sqler/config.yaml` [20 lines] *Default config.yaml of the SQLer agent: sql_connection, script.*
F0690 `Tlamatini/agent/agents/sqler/sqler.py` [217 lines] *conhost.exe orphan guard ——————— When Tlamatini's runtime launches us with DETACHED_PROCESS we have no console...*

SSHer — sheet 20

F0691 `Tlamatini/agent/agents/ssher/config.yaml` [22 lines] *Default config.yaml of the SSHer agent: user, ip, script.*
F0692 `Tlamatini/agent/agents/ssher/ssher.py` [448 lines] *SSHer Agent - Deterministic agent to execute a command via SSH Connects to a remote host and runs the configured script.*

STM32er — sheet 20

F0695 `Tlamatini/agent/agents/stm32er/config.yaml` [242 lines] *Default config.yaml of the STM32er agent: action, stm32_backend, board, framework, environment, server_script, auto_bootstrap +50 more.*
F0696 `Tlamatini/agent/agents/stm32er/stm32er.py` [2.7k lines] *STM32er Agent - STM32 Template Project MCP-server bridge (firmware scaffold/build/flash/observe) Action: Triggered by upstream -> spawn the STM32...*

Summarizer — sheet 20

F0699 `Tlamatini/agent/agents/summarizer/config.yaml` [57 lines] *Default config.yaml of the Summarizer agent: input_text, target_words, system_prompt, llm, poll_interval.*
F0700 `Tlamatini/agent/agents/summarizer/summarizer.py` [613 lines] *Summarizer Agent - Log file monitoring and LLM-powered event detection Action: Started -> Continuously poll source agent logs -> Query LLM with...*

Talker — sheet 20

F0701 `Tlamatini/agent/agents/talker/config.yaml` [151 lines] *Default config.yaml of the Talker agent: input_text, ollama_url, ollama_token, model, language, voice, gender +20 more.*

F0702 `Tlamatini/agent/agents/talker/talker.py` [1.5k lines] *Talker Agent - TEXT-TO-SPEECH (TTS): speaks input_text aloud via OLLAMA.*

Terminater — sheet 20

F0712 `Tlamatini/agent/agents/terminater/config.yaml` [86 lines] *Default config.yaml of the Terminater agent: action, terminal, kind, title, cwd, text, text_b64 +33 more.*
F0713 `Tlamatini/agent/agents/terminater/terminater.py` [726 lines] *Terminater Agent - visible, user-supervised terminals (CMD / PowerShell) Action: raise / type into / read / abort / kill real terminals the USER can...*
F0714 `Tlamatini/agent/agents/terminater/terminater_conpty.py` [667 lines] *terminater_conpty.py – a Windows ConPTY (pseudo console) wrapper in pure ctypes.*
F0715 `Tlamatini/agent/agents/terminater/terminater_engine.py` [1.9k lines] *terminater_engine.py – sessions, the hub and the action API of Terminater.*
F0716 `Tlamatini/agent/agents/terminater/terminater_text.py` [1.1k lines] *terminater_text.py – the pure-text half of Terminater (stdlib only, no I/O).*

Torrenter — sheet 20

F0717 `Tlamatini/agent/agents/torrenter/activity.py` [126 lines] *Human-readable operator log.*
F0718 `Tlamatini/agent/agents/torrenter/client.py` [260 lines] *Transport for the installed Windows Jobs service.*
F0719 `Tlamatini/agent/agents/torrenter/config.yaml` [9 lines] *Default config.yaml of the Torrenter agent: action, arguments, wait_for_completion.*
F0720 `Tlamatini/agent/agents/torrenter/contract.py` [280 lines] *Closed action vocabulary shared by the typed tool, flow nodes and Jobs adapter.*
F0721 `Tlamatini/agent/agents/torrenter/sources.py` [85 lines] *Ephemeral encrypted tracker/web-seed lists.*
F0722 `Tlamatini/agent/agents/torrenter/torrenter.py` [327 lines] *One isolated activation.*

Unrealer — sheet 20

F0723 `Tlamatini/agent/agents/unrealer/config.yaml` [160 lines] *Default config.yaml of the Unrealer agent: host, port, command, params, connect_timeout, read_timeout.*
F0724 `Tlamatini/agent/agents/unrealer/unrealer.py` [873 lines] *Unrealer Agent — drives Unreal Engine via the Unreal MCP plugin's TCP socket protocol (127.0.0.1:55557 by default).*

Video-Analyzer — sheet 20

F0725 `Tlamatini/agent/agents/video_analyzer/config.yaml` [202 lines] *Default config.yaml of the Video-Analyzer agent: video_pathfilenames, analysis_type, audio_tracks, transcription, download, summary_prompt, summary_language +18 more.*
F0726 `Tlamatini/agent/agents/video_analyzer/README.md` [241 lines] *Video-Analyzer: robotics, transcription and summaries: analysis_type selects one of three tasks.*
F0727 `Tlamatini/agent/agents/video_analyzer/video_analyzer.py` [1.2k lines] *Video-Analyzer Agent - robotics verdict, video-track transcription, or summary.*
F0728 `Tlamatini/agent/agents/video_analyzer/video_content.py` [589 lines] *Video audio transcription and evidence-based summaries for standalone pools.*
F0729 `Tlamatini/agent/agents/video_analyzer/video_fetch.py` [855 lines] *Video-Analyzer URL intake: turn a web video link into a local media file.*

VideoPlayer — sheet 20

F0730 `Tlamatini/agent/agents/videoplayer/config.yaml` [69 lines] *Default config.yaml of the VideoPlayer agent: video_file, display_index, volume_percent, time_played, window_width, window_height, fullscreen +1 more.*
F0731 `Tlamatini/agent/agents/videoplayer/videoplayer.py` [974 lines] *VideoPlayer Agent - Plays a video FILE (with audio) on a chosen DISPLAY.*

Whisperer — sheet 21

F0741 `Tlamatini/agent/agents/whisperer/chat_worker.py` [232 lines] *Internal resident dictation worker under source/carried Python.*
F0742 `Tlamatini/agent/agents/whisperer/config.yaml` [182 lines] *Default config.yaml of the Whisperer agent: input_source, audio_file, record_seconds, silence_timeout_seconds, silence_gate, max_record_seconds, silence_threshold_db +26 more.*
F0743 `Tlamatini/agent/agents/whisperer/README.md` [82 lines] *Whisperer: workflow speech recognition and direct chat dictation: Whisperer records the configured host microphone or transcribes audio files.*
F0744 `Tlamatini/agent/agents/whisperer/whisperer.py` [1.7k lines] *Whisperer Agent - Self-contained SPEECH-TO-TEXT (STT / voice recognition) Action: Triggered by upstream OR run from chat -> get audio (either RECORD...*

Windower — sheet 21

F0745 `Tlamatini/agent/agents/windower/config.yaml` [77 lines] *Default config.yaml of the Windower agent: action, window_title, match_mode, match_index, pos_x, pos_y, width +4 more.*
F0746 `Tlamatini/agent/agents/windower/windower.py` [886 lines] *Windower Agent - Application window manager Action: Triggered by upstream -> locate a window by title and run a window lifecycle op (focus / move /...*

Emailer — sheet 21

F0535 `Tlamatini/agent/agents/emailer/config.yaml` [147 lines] *Default config.yaml of the Emailer agent: pattern, poll_interval, smtp, email.*
F0536 `Tlamatini/agent/agents/emailer/emailer.py` [689 lines] *Emailer Agent - No LLM, deterministic email notification agent This agent monitors log files of source agents for a configurable pattern string and...*

FlowHypervisor — sheet 21

F0571 `Tlamatini/agent/agents/flowhypervisor/config.yaml` [18 lines] *Default config.yaml of the FlowHypervisor agent: llm, monitoring_poll_time, user_instructions.*
F0572 `Tlamatini/agent/agents/flowhypervisor/flowhypervisor.py` [786 lines] *FlowHypervisor Agent - LLM-powered flow monitoring agent Watches all running agents' processes and log files, uses an LLM to detect anomalies, and...*
F0573 `Tlamatini/agent/agents/flowhypervisor/hypervisor_alert.wav` [215KB] *Notification sound played by the chat/canvas pages.*
F0574 `Tlamatini/agent/agents/flowhypervisor/monitoring-prompt.pmt` [1.1k lines] *FlowHypervisor watchdog prompt: answers OK or ATTENTION NEEDED, with per-agent special notes and normal-behaviour lists.*

Monitor Netstat — sheet 21

F0620 `Tlamatini/agent/agents/monitor_netstat/config.yaml` [42 lines] *Default config.yaml of the Monitor Netstat agent: llm, target, system_prompt.*
F0621 `Tlamatini/agent/agents/monitor_netstat/monitor_netstat.py` [298 lines] *FIX: Disable Intel Fortran runtime Ctrl+C handler to prevent "forrtl: error (200)"*

Monitor-Log — sheet 21

F0618 `Tlamatini/agent/agents/monitor_log/config.yaml` [44 lines] *Default config.yaml of the Monitor-Log agent: llm, target, system_prompt.*
F0619 `Tlamatini/agent/agents/monitor_log/monitor_log.py` [418 lines] *FIX: Disable Intel Fortran runtime Ctrl+C handler to prevent "forrtl: error (200)"*

Notifier — sheet 21

F0631 `Tlamatini/agent/agents/notifier/config.yaml` [17 lines] *Default config.yaml of the Notifier agent: llm, target.*

F0632 `Tlamatini/agent/agents/notifier/notifier.py` [492 lines] *conhost.exe orphan guard ——————— When Tlamatini's runtime launches us with DETACHED_PROCESS we have no console. . .*

RecMailer — sheet 21

F0677 `Tlamatini/agent/agents/recmailer/config.yaml` [63 lines] *Default config.yaml of the RecMailer agent: poll_interval, max_checks, imap, llm, keywords_or_phrases, outcome_word, system_prompt.*
F0678 `Tlamatini/agent/agents/recmailer/recmailer.py` [367 lines] *FIX: Disable Intel Fortran runtime Ctrl+C handler*

Telegrammer — sheet 21

F0703 `Tlamatini/agent/agents/telegrammer/config.yaml` [27 lines] *Default config.yaml of the Telegrammer agent: mode, telegram, message, contact_name, rx_max_seconds, rx_from_chat_id, rx_match +2 more.*
F0704 `Tlamatini/agent/agents/telegrammer/guide_assets/telegram_01_botfather_search.jpg` [72KB] *Step-by-step setup screenshot used by the messaging-agent guide.*
F0705 `Tlamatini/agent/agents/telegrammer/guide_assets/telegram_02_newbot_token.jpg` [124KB] *Step-by-step setup screenshot used by the messaging-agent guide.*
F0706 `Tlamatini/agent/agents/telegrammer/guide_assets/telegram_03_start_bot_getupdates.jpg` [116KB] *Step-by-step setup screenshot used by the messaging-agent guide.*
F0707 `Tlamatini/agent/agents/telegrammer/guide_assets/telegram_04_tlamatini_contacts.jpg` [119KB] *Step-by-step setup screenshot used by the messaging-agent guide.*
F0708 `Tlamatini/agent/agents/telegrammer/HOW_TO_GET_YOUR_TELEGRAM_ASSETS.md` [303 lines] *Telegrammer Setup: Send A Telegram Message From Tlamatini: This guide is for absolute beginners.*
F0709 `Tlamatini/agent/agents/telegrammer/telegrammer.py` [1.1k lines] *Telegrammer Agent — the ONE Telegram send/receive agent.*

Whatsapper — sheet 21

F0733 `Tlamatini/agent/agents/whatsapper/config.yaml` [62 lines] *Default config.yaml of the Whatsapper agent: mode, provider, whatsapp, message, contact_name, template, template_language +6 more.*
F0734 `Tlamatini/agent/agents/whatsapper/guide_assets/whatsapp_01_meta_create_app.jpg` [90KB] *Step-by-step setup screenshot used by the messaging-agent guide.*
F0735 `Tlamatini/agent/agents/whatsapper/guide_assets/whatsapp_02_api_setup_keys.jpg` [204KB] *Step-by-step setup screenshot used by the messaging-agent guide.*
F0736 `Tlamatini/agent/agents/whatsapper/guide_assets/whatsapp_03_test_recipient_send.jpg` [249KB] *Step-by-step setup screenshot used by the messaging-agent guide.*
F0737 `Tlamatini/agent/agents/whatsapper/guide_assets/whatsapp_04_system_user_token.jpg` [139KB] *Step-by-step setup screenshot used by the messaging-agent guide.*
F0738 `Tlamatini/agent/agents/whatsapper/guide_assets/whatsapp_05_tlamatini_contacts.jpg` [104KB] *Step-by-step setup screenshot used by the messaging-agent guide.*
F0739 `Tlamatini/agent/agents/whatsapper/HOW_TO_GET_YOUR_WHATSAPP_ASSETS.md` [343 lines] *Whatsapper Setup: Send A WhatsApp Message From Tlamatini: This guide is for absolute beginners.*
F0740 `Tlamatini/agent/agents/whatsapper/whatsapper.py` [1.0k lines] *Whatsapper Agent — the ONE WhatsApp send/receive agent.*

FlowBacker — sheet 21

F0563 `Tlamatini/agent/agents/flowbacker/config.yaml` [10 lines] *Default config.yaml of the FlowBacker agent: target_directory.*
F0564 `Tlamatini/agent/agents/flowbacker/flowbacker.py` [468 lines] *FlowBacker Agent - Utility batch infrastructure backup agent Action: Triggered by upstream -> Copy session-id directory to target -> Trigger. . .*

FlowCreator — sheet 21

F0565 `Tlamatini/agent/agents/flowcreator/agentic_skill.md` [2.6k lines] *Tlamatini Flow Creation Skill: You are an expert flow designer for the Tlamatini platform.*
F0566 `Tlamatini/agent/agents/flowcreator/config.yaml` [40 lines] *Default config.yaml of the FlowCreator agent: flow_filename, prompt, output_dir, llm.*
F0567 `Tlamatini/agent/agents/flowcreator/flow_catalog.json` [5.0k lines] *Generated agent catalog (contracts + output fields) read by FlowCreator and Parametrizer.*
F0568 `Tlamatini/agent/agents/flowcreator/flow_knowledge.py` [295 lines] *Standalone contract-aware flow planning helpers; no Django or application imports.*
F0569 `Tlamatini/agent/agents/flowcreator/flowcreator.py` [609 lines] *FlowCreator Agent - Uses an LLM to design and create agent flows automatically Action: Read prompt + agentic_skill.md -> Query LLM -> Parse response. . .*
F0570 `Tlamatini/agent/agents/flowcreator/result_to_flw.py` [236 lines] *result_to_flw.py — convert FlowCreator's flow_result.json into a canonical .flw flow file that the Tlamatini ACP canvas loader (loadDiagram in. . .*

Parametrizer — sheet 21

F0635 `Tlamatini/agent/agents/parametrizer/config.yaml` [11 lines] *Default config.yaml of the Parametrizer agent: source_agent, target_agent.*
F0636 `Tlamatini/agent/agents/parametrizer/parametrizer.py` [1.2k lines] *Parametrizer Agent - Utility Interconnection Agent Maps structured outputs from a source agent's log to a target agent's config.yaml Reads. . .*

Kyber-Cipher — sheet 21

F0601 `Tlamatini/agent/agents/kyber_cipher/config.yaml` [14 lines] *Default config.yaml of the Kyber-Cipher agent: kyber_variant, public_key, buffer.*
F0602 `Tlamatini/agent/agents/kyber_cipher/kyber_cipher.py` [580 lines] *Kyber-Cipher Agent - Encrypts a buffer using a CRYSTALS-Kyber public key*

Kyber-DeCipher — sheet 21

F0603 `Tlamatini/agent/agents/kyber_decipher/config.yaml` [20 lines] *Default config.yaml of the Kyber-DeCipher agent: kyber_variant, private_key, encapsulation, initialization_vector, cipher_text.*
F0604 `Tlamatini/agent/agents/kyber_decipher/kyber_decipher.py` [519 lines] *Kyber-DeCipher Agent - Decrypts a cipher text using a CRYSTALS-Kyber private key*

Kyber-KeyGen — sheet 21

F0605 `Tlamatini/agent/agents/kyber_keygen/config.yaml` [8 lines] *Default config.yaml of the Kyber-KeyGen agent: kyber_variant.*
F0606 `Tlamatini/agent/agents/kyber_keygen/kyber_keygen.py` [483 lines] *Kyber-KeyGen Agent - Generates CRYSTALS-Kyber public/private key pairs*

TeleTlamatini — sheet 21

F0710 `Tlamatini/agent/agents/teletlamatini/config.yaml` [101 lines] *Default config.yaml of the TeleTlamatini agent: telegram, password, tlamatini, completeness_check.*
F0711 `Tlamatini/agent/agents/teletlamatini/teletlamatini.py` [1.3k lines] *TeleTlamatini Agent — pure Telegram-bot bridge to the Tlamatini chat core.*

ACPXer — sheet 21

F0490 `Tlamatini/agent/agents/acpxer/acpxer.py` [914 lines] *ACPXer Agent - Visual ACPX Session Driver*
F0491 `Tlamatini/agent/agents/acpxer/config.yaml` [43 lines] *Default config.yaml of the ACPXer agent: agent_id, command, task, mode, cwd, idle_seconds, timeout_seconds +1 more.*

Atlas VII — Verification: unit, contract, visible and release tests

agent/ · unit and contract tests — sheet 22

F1396 `Tlamatini/agent/test_access_key_wizard.py` [145 lines] *Tests for the graphical Access Keys Wizard backend. [2 tests]*
F1397 `Tlamatini/agent/test_agent_display_names.py` [287 lines] *Pins the AGENT DISPLAY-NAME contract (Angela, 2026-07-26). [11 tests]*
F1398 `Tlamatini/agent/test_agent_manifest.py` [449 lines] *Tests for the companion-app discovery surface (Tlamatini-FlowPills PROP-001. . . 003) [19 tests]*
F1399 `Tlamatini/agent/test_agent_verdict.py` [293 lines] *Tests for the deterministic verdict engine (agent/agent_verdict.py). [25 tests]*
F1400 `Tlamatini/agent/test_agents_see_hidden_dirs.py` [71 lines] *The file agents must see dot-directories, and still spare .git. [4 tests]*
F1401 `Tlamatini/agent/test_arduiner_agent.py` [717 lines] *Automated tests for the Arduiner workflow agent and its surrounding infrastructure. [41 tests]*
F1402 `Tlamatini/agent/test_ask_execs_allowlist.py` [230 lines] *Pins ANGELA'S Ask-Execs policy (2026-07-14). [8 tests]*
F1403 `Tlamatini/agent/test_audioplayer_agent.py` [780 lines] *Automated tests for the AudioPlayer workflow agent and its surrounding infrastructure. [44 tests]*
F1404 `Tlamatini/agent/test_binary_guard.py` [445 lines] *Coverage for the binary-content guard on the RAG context-loading chain. [46 tests]*
F1405 `Tlamatini/agent/test_blenderer_agent.py` [577 lines] *Automated tests for the Blenderer workflow agent and its surrounding infrastructure. [40 tests]*
F1406 `Tlamatini/agent/test_build_env.py` [151 lines] *Libraries installed on the build machine must never reach a release (2026-10-21). [13 tests]*
F1407 `Tlamatini/agent/test_build_pip_quiet.py` [206 lines] *Guard: the build pipeline must NEVER print pip's self-upgrade nag. [4 tests]*
F1408 `Tlamatini/agent/test_build_root_assets.py` [229 lines] *Physical packaging regressions; run in a verified visible foreground console. [17 tests]*
F1409 `Tlamatini/agent/test_build_scripts.py` [720 lines] *Static completeness tests for the packaging pipeline — build.py, build_installer.py, build_uninstaller.py. [57 tests]*
F1410 `Tlamatini/agent/test_camcorder_agent.py` [467 lines] *Automated tests for the Camcorder workflow agent and its surrounding infrastructure. [22 tests]*
F1411 `Tlamatini/agent/test_cancellation.py` [215 lines] *Regression tests for the PER-RUN cancellation latch (agent/cancellation.py). [22 tests]*
F1412 `Tlamatini/agent/test_canvas_file_views.py` [348 lines] *Native-save safety tests; run from a verified visible development console. [28 tests]*
F1413 `Tlamatini/agent/test_catalog_pruning.py` [49 lines] *Catalog retirement must work on both new databases and existing installs. [2 tests]*
F1414 `Tlamatini/agent/test_chain_readiness.py` [327 lines] *Regression pins for the PERMANENT CHAT WEDGE (found 2026-07-29, Angela). [15 tests]*
F1415 `Tlamatini/agent/test_chat_agent_run_wait.py` [200 lines] *Regression guard: chat_agent_run_wait must return AS SOON AS the child exits. [5 tests]*
F1416 `Tlamatini/agent/test_chat_bridge_bots.py` [234 lines] *Tests for the long-running chat-bridge bot — TeleTlamatini — focused on the post-"Ask Execs" adaptation (2026-06-04). [9 tests]*
F1417 `Tlamatini/agent/test_chat_history_window.py` [94 lines] *Regression tests for the conversation-memory window (chat_history_loader). [5 tests]*
F1418 `Tlamatini/agent/test_chat_voice.py` [492 lines] *Direct dictation regressions. [36 tests]*
F1419 `Tlamatini/agent/test_checkbox_bulk_toggle.py` [189 lines] *Guards the SPACE-bar bulk check/uncheck of selected checkboxes. [15 tests]*
F1420 `Tlamatini/agent/test_command_watchdog.py` [367 lines] *Hard, real-scenario tests for agent/command_watchdog.py. [15 tests]*
F1421 `Tlamatini/agent/test_compact_mode_pingpong.py` [117 lines] *Two chat pages on two models must not ping-pong (Angela, 2026-10-08). [4 tests]*
F1422 `Tlamatini/agent/test_compact_mode_switch.py` [468 lines] *The COMPACT MODE switch (Angela, 2026-10-02). [33 tests]*
F1423 `Tlamatini/agent/test_configure_dialog_initial_ticks.py` [54 lines] *Configure Agents / Configure MCPs start from the rows the server sent. [5 tests]*
F1424 `Tlamatini/agent/test_console_colors.py` [517 lines] *CONSOLE COLOURS - the window is painted by level, the log file never is (Angela, 2026-10-08). [33 tests]*
F1425 `Tlamatini/agent/test_console_shield.py` [602 lines] *Contract tests for the CONSOLE SHIELD (Angela, 2026-09-10). [28 tests]*
F1426 `Tlamatini/agent/test_contacts.py` [153 lines] *Tests for the Contacts book (agent/contacts.py) — name -> messaging handle. [11 tests]*
F1427 `Tlamatini/agent/test_context_governor.py` [572 lines] *Tests for the Context Governor — STEP 1, measurement only. [52 tests]*
F1428 `Tlamatini/agent/test_context_real_tokens.py` [691 lines] *Tests for the REAL context gauge (Angela, 2026-09-28). [46 tests]*
F1429 `Tlamatini/agent/test_context_restore_spinner.py` [204 lines] *A RESTORED CONTEXT MUST LOOK AS BUSY AS IT IS (2026-09-22). [5 tests]*
F1430 `Tlamatini/agent/test_context_sidecar_timeout.py` [258 lines] *The context sidecars' own Ollama calls are bounded (Angela, 2026-10-05). [16 tests]*
F1431 `Tlamatini/agent/test_crawler_agent.py` [342 lines] *Tests for the Crawler agent's recon / multi-seed / safety improvements. [42 tests]*
F1432 `Tlamatini/agent/test_crawler_resilience.py` [503 lines] *The Crawler must read the page it was given, never hang, and never lie (2026-09-28). [40 tests]*
F1433 `Tlamatini/agent/test_createsuperuser_wizard.py` [71 lines] *Tests for the create-superuser wizard catalog prompt (migration 0145). [4 tests]*
F1434 `Tlamatini/agent/test_ctrl_c_shutdown.py` [216 lines] *Ctrl+C MUST ALWAYS QUIT — source contract for the shutdown path. [11 tests]*
F1435 `Tlamatini/agent/test_db_backup_restore_wal.py` [432 lines] *DB ▷ Backup database and DB ▷ Set DB must work under WAL. [24 tests]*
F1436 `Tlamatini/agent/test_db_backup_safety.py` [130 lines] *The daily DB backup must refuse bad input and must never fail quietly. [7 tests]*
F1437 `Tlamatini/agent/test_de_compresser.py` [843 lines] *Automated tests for the De-Compresser workflow agent. [44 tests]*
F1438 `Tlamatini/agent/test_deep_research_prompt.py` [86 lines] *Tests for the Deep Internet Research catalog prompt (migration 0194). [4 tests]*
F1439 `Tlamatini/agent/test_deleter_safety.py` [128 lines] *Guard the Deleter against ever deleting the directory it was told to work IN. [9 tests]*
F1440 `Tlamatini/agent/test_desktop_input_agents.py` [366 lines] *Desktop-input regressions. [38 tests]*
F1441 `Tlamatini/agent/test_dialog_dismissal_policy.py` [567 lines] *DIALOG DISMISSAL POLICY - COVERAGE AUDIT (Angela, 2026-08-13) [35 tests]*
F1442 `Tlamatini/agent/test_dialog_theme.py` [767 lines] *DIALOG STANDARDIZATION CONTRACT (Angela, 2026-08-12) [42 tests]*
F1443 `Tlamatini/agent/test_discoverer_agent.py` [533 lines] *Automated tests for the Discoverer workflow agent and its surrounding wiring. [40 tests]*
F1444 `Tlamatini/agent/test_discoverer_thousand.py` [141 lines] *1000 data-driven tests for the Discoverer agent, runnable in the Tlamatini dev (python manage.py test agent.test_discoverer_thousand).*
F1445 `Tlamatini/agent/test_django_port_config.py` [254 lines] *Contract tests for the CONFIGURABLE WEB PORT — config.json → django_port. [24 tests]*
F1446 `Tlamatini/agent/test_drag_divider_look.py` [70 lines] *drag-divider has the same look and feel as #vertical-drag-divider (2026-10-11). [5 tests]*
F1447 `Tlamatini/agent/test_drop_message.py` [331 lines] *The chat card's DROP button (Angela Lopez Mendoza, 2026-09-30). [23 tests]*
F1448 `Tlamatini/agent/test_editor_agent.py` [181 lines] *HARD real-scenario tests for the Editor agent (surgical in-place find/replace). [15 tests]*
F1449 `Tlamatini/agent/test_embedding_memory_guard.py` [638 lines] *Tests for agent.embedding_memory_guard. [49 tests]*
F1450 `Tlamatini/agent/test_esp32er_agent.py` [767 lines] *Automated tests for the ESP32er workflow agent and its surrounding infrastructure. [43 tests]*
F1451 `Tlamatini/agent/test_esphomer_agent.py` [554 lines] *Automated tests for the ESPHomer workflow agent and its surrounding infrastructure. [41 tests]*
F1452 `Tlamatini/agent/test_exec_report_verdict.py` [142 lines] *Regression guards for the Exec Report SUCCESS/FAILURE verdict. [19 tests]*
F1453 `Tlamatini/agent/test_execute_command_bounded.py` [176 lines] *Hard regression tests for the bounded execute_command (2026-06-06). [8 tests]*
F1454 `Tlamatini/agent/test_executer_shell_request.py` [34 lines] *Keep CMD environment assignments inside Executer's shell script. [3 tests]*
F1455 `Tlamatini/agent/test_executer_window_report.py` [148 lines] *Executer must report its forked window TRUTHFULLY (2026-10-09). [11 tests]*
F1456 `Tlamatini/agent/test_external_mcp_add_flow.py` [179 lines] *Certify the fix for: "I don't have a file-writing or shell tool bound this turn" when the user asks to add an MCP (e.g. [12 tests]*
F1457 `Tlamatini/agent/test_external_mcp_e2e.py` [470 lines] *End-to-end certification of the External MCPs pipeline: from the dropped JSON all the way to executing, diagnosing, and supervising a live MCP server. [10 tests]*
F1458 `Tlamatini/agent/test_external_mcp_transports.py` [294 lines] *End-to-end loopback tests for the External MCP network transports. [11 tests]*
F1459 `Tlamatini/agent/test_external_mcp_universal.py` [674 lines] *External MCP universal hardening tests. [39 tests]*
F1460 `Tlamatini/agent/test_file_creator_append.py` [241 lines] *Deterministic proof for File-Creator's append option (Angela, 2026-10-08). [17 tests]*
F1461 `Tlamatini/agent/test_file_drop_router.py` [251 lines] *Guards the page-wide file-drop router (Angela, 2026-10-14). [22 tests]*
F1462 `Tlamatini/agent/test_file_extractor_reader.py` [104 lines] *HARD tests for the File-Extractor Reader upgrade (line_numbers + offset/limit). [6 tests]*
F1464 `Tlamatini/agent/test_flow_contracts.py` [227 lines] *Defines FlowContractTests. [10 tests]*
F1465 `Tlamatini/agent/test_flow_file_open.py` [160 lines] *Run from a verified visible foreground console; no hidden test execution. [13 tests]*
F1466 `Tlamatini/agent/test_flow_file_views.py` [100 lines] *Run via manage.py test in a verified visible foreground console. [7 tests]*
F1467 `Tlamatini/agent/test_flow_knowledge.py` [369 lines] *Offline flow/catalog regressions. [33 tests]*
F1468 `Tlamatini/agent/test_frontend_mutable_state.py` [260 lines] *Defines FrontendMutableStateTests, PromptCatalogEndpointTests. [8 tests]*
F1469 `Tlamatini/agent/test_gitter_custom_command.py` [228 lines] *Gitter custom_command tokenization — Windows paths must survive. [23 tests]*
F1470 `Tlamatini/agent/test_glob_patterns.py` [508 lines] *Brace sets, pattern lists and folder globs for Grepper and Globber (2026-10-18). [47 tests]*
F1471 `Tlamatini/agent/test_globber_agent.py` [156 lines] *HARD real-scenario tests for the Globber agent (read-only filename glob search). [13 tests]*
F1472 `Tlamatini/agent/test_googler_agent.py` [502 lines] *Tests for the Googler agent's dork-search improvements. [41 tests]*
F1473 `Tlamatini/agent/test_googler_dorks.py` [513 lines] *Tests for Googler's Google-dork query builder (2026-08-23). [73 tests]*
F1474 `Tlamatini/agent/test_googler_resilience.py` [800 lines] *Googler must never hang, never lie, and never hammer (2026-09-28). [65 tests]*
F1475 `Tlamatini/agent/test_grepper_agent.py` [172 lines] *HARD real-scenario tests for the Grepper agent (read-only regex content search). [15 tests]*
F1476 `Tlamatini/agent/test_grepper_encodings.py` [222 lines] *Grepper must not go BLIND on non-UTF-8 text. [16 tests]*
F1477 `Tlamatini/agent/test_grepper_lines_mode.py` [279 lines] *HARD real-scenario tests for Grepper's output_mode: "lines" VERBATIM READ. [21 tests]*
F1478 `Tlamatini/agent/test_image_interpreter_agent.py` [401 lines] *Hard tests for the Image-Interpreter TRIPLE-MODEL pipeline (2026-07-04). [18 tests]*
F1479 `Tlamatini/agent/test_instant_messaging_doctor.py` [171 lines] *Instant Messaging Doctor contract and Parametrizer integration tests. [5 tests]*
F1480 `Tlamatini/agent/test_interaction_settings.py` [196 lines] *Run from a verified foreground PowerShell -NoExit console. [15 tests]*
F1481 `Tlamatini/agent/test_interactor.py` [319 lines] *Interactor protocol, ownership, cancellation and persistence regressions. [26 tests]*
F1482 `Tlamatini/agent/test_j_decompiler_agent.py` [252 lines] *Automated tests for the J-Decompiler agent + the decompile_java tool. [10 tests]*
F1483 `Tlamatini/agent/test_js_gates.py` [60 lines] *Process gates for the browser JavaScript – two silent, total failure modes. [2 tests]*
F1484 `Tlamatini/agent/test_kalier_agent.py` [1.1k lines] *Automated tests for the Kalier workflow agent and its surrounding infrastructure. [83 tests]*
F1485 `Tlamatini/agent/test_latexer_agent.py` [1.3k lines] *Automated tests for the LaTeXer workflow agent and its surrounding infrastructure. [100 tests]*
F1486 `Tlamatini/agent/test_latexer_delivery_truth.py` [66 lines] *LaTeXer must verify the delivered file before reporting a finished PDF. [4 tests]*
F1487 `Tlamatini/agent/test_latexer_layout_truth.py` [60 lines] *Large overfull TeX boxes must never receive the finished-document verdict. [5 tests]*
F1488 `Tlamatini/agent/test_latexer_major_removal_and_window_repair.py` [464 lines] *2026-10-08 – two fixes born from Angela's Podman cheat-sheet run (46.6 minutes). [37 tests]*
F1489 `Tlamatini/agent/test_latexer_model_repair_truth.py` [55 lines] *A repair must typeset the document, not print its source as a code listing. [3 tests]*
F1490 `Tlamatini/agent/test_latexer_repair_ladder.py` [539 lines] *Regression guards for the LaTeXer REPAIR LADDER (Layer 05). [48 tests]*
F1491 `Tlamatini/agent/test_latexer_styles.py` [186 lines] *Design invariants, caller contracts and copied-pool execution. [15 tests]*
F1492 `Tlamatini/agent/test_latexer_suite.py` [1.7k lines] *LaTeXer — the BIG suite: a fixed, permanent testing asset (150+ tests). [318 tests]*
F1493 `Tlamatini/agent/test_latexer_verbatim_channel.py` [326 lines] *Regression tests for the LaTeXer BYTE-EXACT (verbatim + base64) channel. [19 tests]*
F1494 `Tlamatini/agent/test_log_identity.py` [349 lines] *Behavioural tests for per-line USER attribution in tlamatini.log. [28 tests]*
F1495 `Tlamatini/agent/test_mcp_configured_endpoints.py` [217 lines] *Real nondefault MCP listeners and their configured clients, without inference. [8 tests]*
F1496 `Tlamatini/agent/test_model_brain.py` [511 lines] *MODEL BRAIN - every decision is formal, per model, and fail-open (Angela, 2026-10-08). [42 tests]*
F1497 `Tlamatini/agent/test_model_capacity.py` [456 lines] *COMPACT MODE - every request fits the model it is sent to (Angela, 2026-10-01). [46 tests]*
F1498 `Tlamatini/agent/test_model_settings.py` [221 lines] *Model selection across UI, standalone agents and isolated runtimes. [11 tests]*
F1499 `Tlamatini/agent/test_mover_file_safety.py` [218 lines] *Regression for the 2026-10-07 scalar-path Desktop deletion incident. [18 tests]*
F1500 `Tlamatini/agent/test_native_dialogs.py` [294 lines] *Tests for the tkinter-free native file/folder picker. [26 tests]*
F1501 `Tlamatini/agent/test_netspeed_calculator_agent.py` [1.6k lines] *Hard, real-scenario tests for the NetSpeed-Calculator agent (#88). [262 tests]*
F1502 `Tlamatini/agent/test_nmapper_agent.py` [268 lines] *Hard, real-code tests for the Nmapper agent (LOCAL, use-only nmap bridge) and its registration across the Tlamatini surfaces. [28 tests]*
F1503 `Tlamatini/agent/test_no_audio_in_tests.py` [60 lines] *A test run must NEVER make a sound on the developer's desktop. [4 tests]*
F1504 `Tlamatini/agent/test_ollama_catalog_from_server.py` [282 lines] *The Config -> Models dialog must ask the CONFIGURED Ollama, not the local one. [15 tests]*
F1505 `Tlamatini/agent/test_ollama_stop_limit.py` [250 lines] *No Ollama request may carry more than four stop sequences, and a failed chat-history summary must never fail the answer (Angela, 2026-10-03). [16 tests]*
F1506 `Tlamatini/agent/test_orphan_reaper.py` [340 lines] *Regression tests for agent.orphan_reaper. [25 tests]*
F1507 `Tlamatini/agent/test_parametrizer_mcp_doctor.py` [145 lines] *Verify the Parametrizer change Codex made for MCP Doctor works end to end. [4 tests]*
F1508 `Tlamatini/agent/test_password_quoting.py` [605 lines] *Password-quoting contract tests for Emailer and Recmailer. [28 tests]*
F1509 `Tlamatini/agent/test_pdf_context.py` [307 lines] *PDF upload permissions, extraction, user isolation and real consumer handoff. [17 tests]*
F1510 `Tlamatini/agent/test_pdfer_agent.py` [1.2k lines] *Automated tests for the PDFer workflow agent and its surrounding infrastructure. [82 tests]*
F1511 `Tlamatini/agent/test_pdfer_nuance_layout.py` [654 lines] *Regression guard for the PDFer nuance / typography / layout overhaul. [37 tests]*
F1512 `Tlamatini/agent/test_pdfer_styles.py` [192 lines] *Rendered regressions for PDFer's visual styles and protected reading areas. [11 tests]*
F1513 `Tlamatini/agent/test_playwrighter_agent.py` [662 lines] *Automated tests for the Playwrighter workflow agent. [60 tests]*
F1514 `Tlamatini/agent/test_pptxer_agent.py` [1.3k lines] *PPTXer — the agent that MEASURES ITS OWN SLIDES, and the tests that measure it. [96 tests]*
F1515 `Tlamatini/agent/test_pptxer_styles.py` [113 lines] *Style selection, user overrides, and long-content layout regressions. [7 tests]*
F1516 `Tlamatini/agent/test_pptxer_visibility.py` [308 lines] *Regression tests for real PPTX content preservation and bounded text. [23 tests]*
F1517 `Tlamatini/agent/test_preserved_user_state.py` [201 lines] *USER STATE MUST SURVIVE A REINSTALL (Angela, 2026-08-12) [10 tests]*
F1518 `Tlamatini/agent/test_program_name_collisions.py` [187 lines] *Two code blocks in ONE answer must never break each other's file. [11 tests]*
F1519 `Tlamatini/agent/test_prompt_catalog_contiguous.py` [311 lines] *Catalog-of-Prompts INVARIANTS. [15 tests]*
F1520 `Tlamatini/agent/test_prompt_flow_panel.py` [292 lines] *Run in a visible foreground console: python Tlamatini/agent/test_prompt_flow_panel.py. [29 tests]*
F1521 `Tlamatini/agent/test_prompt_flow_panel_carriage.py` [96 lines] *Packaging omission regressions; run in a verified visible foreground console. [3 tests]*
F1522 `Tlamatini/agent/test_prompt_flow_panel_runtime.py` [198 lines] *Run in a verified visible foreground console with -NoExit: python Tlamatini/manage.py test agent.test_prompt_flow_panel_runtime –noinput Adapter. . . [9 tests]*
F1523 `Tlamatini/agent/test_prompt_flow_panel_websocket.py` [173 lines] *Real ASGI routing/consumer/runner tests; model adapter is deterministic. [10 tests]*
F1524 `Tlamatini/agent/test_prompt_shape_sentences.py` [67 lines] *Offline prompt-shape regressions; no model or filesystem access is invoked. [6 tests]*
F1525 `Tlamatini/agent/test_public_release_targets.py` [486 lines] *test_public_release_targets.py – .private_targets.json is OPTIONAL and is NEVER a runtime dependency. [26 tests]*
F1526 `Tlamatini/agent/test_pyinstaller_guard.py` [149 lines] *The release build must not die on the build machine's clutter (2026-10-21). [9 tests]*
F1527 `Tlamatini/agent/test_recorder_agent.py` [617 lines] *Automated tests for the Recorder workflow agent and its surrounding infrastructure. [37 tests]*
F1528 `Tlamatini/agent/test_release_payload_ownership.py` [173 lines] *Packaging regressions; run in a visible foreground development console. [6 tests]*
F1529 `Tlamatini/agent/test_runtime_provisioner.py` [498 lines] *Behavioural guards for the Runtime Provisioner and the shipped default MCPs. [46 tests]*
F1530 `Tlamatini/agent/test_security_assets_carriage.py` [140 lines] *Guard the Blue-hat security toolkit's CARRIAGE + EVIDENCE-carryover contracts. [15 tests]*
F1531 `Tlamatini/agent/test_self_healing.py` [259 lines] *Deterministic proof for the self-healing, NEVER-HANGING multi-turn redesign (Angela, 2026-07-06). [12 tests]*
F1532 `Tlamatini/agent/test_self_knowledge_is_current.py` [112 lines] *Keep agent/Tlamatini.md (the injected {self_knowledge} system-prompt payload) from drifting behind source. [6 tests]*
F1533 `Tlamatini/agent/test_self_modify_gate.py` [459 lines] *test_self_modify_gate.py — "her source and her self-knowledge ship together, or not at all", and the DEFAULT build carries neither. [28 tests]*
F1534 `Tlamatini/agent/test_self_modify_switch.py` [481 lines] *The SELF-MODIFY switch (Angela, 2026-10-03). [35 tests]*
F1535 `Tlamatini/agent/test_self_update.py` [157 lines] *Tests for the self-update module (About ▷ Check for updates). [17 tests]*
F1536 `Tlamatini/agent/test_shoter_takes_screenshots.py` [72 lines] *GUARDA: en las pruebas, las fotos las toma SHOTER — nunca PIL.ImageGrab. [3 tests]*
F1537 `Tlamatini/agent/test_sqlite_wal_mode.py` [52 lines] *SQLite must stay in WAL mode. [3 tests]*
F1538 `Tlamatini/agent/test_status_vocabulary.py` [570 lines] *THE GUARD that ties the status vocabulary to the agents that emit it. [27 tests]*
F1539 `Tlamatini/agent/test_step_by_step_mode.py` [209 lines] *Deep certification of the Step-by-Step toolbar mode (step_by_step_enabled). [15 tests]*
F1540 `Tlamatini/agent/test_stm32er_agent.py` [1.8k lines] *Automated tests for the STM32er workflow agent and its surrounding infrastructure. [126 tests]*
F1541 `Tlamatini/agent/test_talker_agent.py` [1.3k lines] *Automated tests for the Talker workflow agent and its surrounding infrastructure. [78 tests]*
F1542 `Tlamatini/agent/test_telegrammer_identity.py` [214 lines] *Tests for the Telegrammer dual-identity ("send as me" vs "send as the bot") provider selection. [19 tests]*
F1543 `Tlamatini/agent/test_temp_dir_policy.py` [479 lines] *test_temp_dir_policy.py — the "temp stays inside Tlamatini" contract. [33 tests]*
F1544 `Tlamatini/agent/test_terminater_agent.py` [700 lines] *The terminater pool agent, exactly as a flow or the chat launches it. [70 tests]*
F1545 `Tlamatini/agent/test_terminater_engine.py` [1.7k lines] *Terminater engine: the pure logic AND the real shells it hosts. [205 tests]*
F1546 `Tlamatini/agent/test_terminater_hub_api.py` [846 lines] *Terminater's process-wide hub, its loopback HTTP API and its WebSocket. [110 tests]*
F1547 `Tlamatini/agent/test_terminater_integration.py` [727 lines] *Terminater, wired into the rest of Tlamatini. [117 tests]*
F1548 `Tlamatini/agent/test_terminater_pfp.py` [807 lines] *The Terminater step (the parallelogram) inside the Prompt Flow Panel. [81 tests]*
F1549 `Tlamatini/agent/test_terminater_regressions.py` [269 lines] *Deterministic scheduling/output regressions found by the October 20 audit. [20 tests]*
F1550 `Tlamatini/agent/test_terminater_reliability.py` [1.4k lines] *Terminater reliability: the chat model's door to the terminal can no longer lie, guess or glue. [166 tests]*
F1551 `Tlamatini/agent/test_terminater_text.py` [583 lines] *Terminater's pure text engine: the typist, the key table, the virtual screen, the user's line tracker and the advisor. [108 tests]*
F1552 `Tlamatini/agent/test_tool_failure_recovery.py` [87 lines] *Unit tests for the universal tool-failure classifier + self-correction. [20 tests]*
F1553 `Tlamatini/agent/test_torrenter.py` [364 lines] *Defines ActionContractTests, AccountTests, RecoveryTests. [20 tests]*
F1554 `Tlamatini/agent/test_torrenter_delivery.py` [82 lines] *Regression checks for internal IPC and honest delivered-file reporting. [6 tests]*
F1555 `Tlamatini/agent/test_uninstall_processes.py` [129 lines] *Actual owned-worker cleanup, scoped to a temporary installation. [5 tests]*
F1556 `Tlamatini/agent/test_uninstaller_lifecycle.py` [295 lines] *Uninstall regressions; run in a verified visible foreground console. [21 tests]*
F1557 `Tlamatini/agent/test_uninstaller_mechanics.py` [672 lines] *Guards for the Tlamatini Uninstaller's two safety mechanics (2026-09-13). [56 tests]*
F1558 `Tlamatini/agent/test_unrealer_agent.py` [518 lines] *Automated tests for the Unrealer workflow agent and its surrounding infrastructure. [46 tests]*
F1559 `Tlamatini/agent/test_usage.py` [518 lines] *Usage correctness, isolation, failure handling and source-carriage contracts. [35 tests]*
F1560 `Tlamatini/agent/test_version_guard.py` [62 lines] *Pin the ENGLISH version-metadata + update-detection expectations. [9 tests]*
F1561 `Tlamatini/agent/test_video_analyzer_agent.py` [357 lines] *Hard, real-scenario tests for the Video-Analyzer agent (#84). [29 tests]*
F1562 `Tlamatini/agent/test_video_analyzer_content.py` [459 lines] *Real media decode/sampling plus isolated ASR/LLM integration regressions. [26 tests]*
F1563 `Tlamatini/agent/test_video_analyzer_fetch.py` [595 lines] *Video-Analyzer web links: video_fetch.py, the chat request parser and carriage. [54 tests]*
F1564 `Tlamatini/agent/test_videoplayer_agent.py` [758 lines] *Automated tests for the VideoPlayer workflow agent and its surrounding infrastructure. [38 tests]*
F1565 `Tlamatini/agent/test_visual_fatal_errors.py` [246 lines] *Fatal analysis reporting must remain independent of Tlamatini recovery. [17 tests]*
F1566 `Tlamatini/agent/test_voice_commands_catalog.py` [266 lines] *VOICE COMMANDS — the catalog section added by migration 0201 (2026-09-12). [18 tests]*

F1567 `Tlamatini/agent/test_watchdog_foreground_exemption.py` [672 lines] *Complete automated tests for the generalized "eagle-eye" foreground-console exemption shared by command_watchdog and orphan_reaper. [28 tests]*
F1568 `Tlamatini/agent/test_web_process_stays_lean.py` [232 lines] *The frozen Django process must NOT drag in the transformers/torch ML stack. [10 tests]*
F1569 `Tlamatini/agent/test_whatsapper_identity.py` [202 lines] *Tests for the Whatsapper dual-identity ("send as me" vs the official business number) provider selection. [15 tests]*
F1570 `Tlamatini/agent/test_whisperer_agent.py` [1.1k lines] *Automated tests for the Whisperer workflow agent and its surrounding infrastructure. [72 tests]*
F1571 `Tlamatini/agent/test_win_shim.py` [404 lines] *Every Windows .cmd / .exe spawn scenario Tlamatini has to survive. [27 tests]*
F1572 `Tlamatini/agent/test_windower_agent.py` [662 lines] *Automated tests for the Windower workflow agent. [56 tests]*
F1573 `Tlamatini/agent/test_windows_app_registration.py` [107 lines] *Tests for agent/windows_app_registration.py (Installed-apps / ARP entry). [8 tests]*
F1574 `Tlamatini/agent/test_wizard_intent_guards.py` [219 lines] *User input in a setup wizard must not become command takeover or notification consent. [11 tests]*
F1575 `Tlamatini/agent/tests.py` [7.8k lines] *Defines P0HardeningTests, P1HardeningTests, SetDirectoryAsContextPathTests, ContextDirectoryPickerViewTests and 79 more. [437 tests]*
F1576 `Tlamatini/agent/tests_perf_3x.py` [432 lines] *Automated regression tests for the 3X-speed surgical plan — L1 (Ollama serving layer) + L2 (orphan reaper). [24 tests]*

Tests/ · avatar, PDF canvas and requirement tests — sheet 22

F0445 `Tests/avatar_dev_server/dev_settings.py` [29 lines] *Isolated settings for the requested visible Django avatar tests only.*
F0446 `Tests/avatar_dev_server/development_test.py` [51 lines] *Real development Django server with an isolated database and no agent jobs.*
F0447 `Tests/run_avatar_tests.py` [129 lines] *One-command, VISIBLE avatar regression for the real development Django app.*
F0448 `Tests/run_presence_tests.ps1` [76 lines] *ONE COMMAND, ONE VISIBLE WINDOW, ONE REPORT.*
F0449 `Tests/test_about_window_name_visible.py` [66 lines] *Playwright test — Angela's name is visible in Tlamatini's About window.*
F0450 `Tests/test_avatar_animation.cjs` [196 lines] *Browser regression: run with node tests/test_avatar_animation.cjs.*
F0451 `Tests/test_avatar_collectstatic.py` [66 lines] *Verify real Django collectstatic preserves avatar assets byte-for-byte.*
F0452 `Tests/test_avatar_presence_visible.py` [582 lines] *THE AVATAR PRESENCE REGRESSION — VISIBLE, PERMANENT, AND HARD TO FOOL.*
F0453 `Tests/test_avatar_visible.cjs` [170 lines] *Visible real-Django avatar regression.*
F0454 `Tests/test_build_env_concurrency.py` [160 lines] *Real Python-cache lifecycle tests; run from a visible foreground console. [5 tests]*
F0455 `Tests/test_canvas_text_decoding.cjs` [87 lines] *Run with node –test Tests/test_canvas_text_decoding.cjs from a visible console.*
F0456 `Tests/test_pdf_canvas_assets.py` [68 lines] *Check real Django/WhiteNoise MIME handling and collected PDF assets in isolation. [1 tests]*
F0457 `Tests/test_pdf_canvas_browser.py` [738 lines] *Real Chromium PDF canvas regression, isolated from RAG, accounts and databases. [11 tests]*
F0458 `Tests/test_perf_3x_visual.py` [216 lines] *VISUAL automated tests for the 3X-speed surgical plan (L1 + L2).*
F0459 `Tests/test_release_regeneration.py` [270 lines] *Release regeneration must preserve credential values and contact updates. [19 tests]*
F0460 `Tests/test_release_type_banner.py` [143 lines] *Release metadata and real log-tee tests without Django/server startup. [5 tests]*
F0461 `Tests/test_release_version_pinning.py` [164 lines] *Release version regressions; run in a verified visible foreground console. [6 tests]*
F0462 `Tests/test_requirements_coverage.py` [103 lines] *Run visibly: python -m unittest discover -s Tests -p test_requirements_coverage.py -v [12 tests]*
F0463 `Tests/test_setup_release_paths.py` [627 lines] *Real filesystem setup regressions; run in a visible foreground console. [34 tests]*
F0464 `Tests/test_update_client_carriage.py` [88 lines] *Native filesystem checks of the actual update helpers, run in a visible console. [2 tests]*

AuxTests/ — sheet 22

F0212 `AuxTests/mypdf2.py` [156 lines] *Defines create_beautiful_pdf().*

Tlamatini/tests_e2e/ · visible end-to-end runs — sheet 22

F1604 `Tlamatini/tests_e2e/oob_child_helper.py` [47 lines] *Child process used by the OOB_shift_reaper unit test.*
F1605 `Tlamatini/tests_e2e/README.md` [79 lines] *tests_e2e — visible (headed) Playwright end-to-end suites: These drive a real, visible Chromium against a running Tlamatini server (they are NOT Django unit…*
F1606 `Tlamatini/tests_e2e/test_create_flow_visual.py` [477 lines] *Visible (HEADED) Playwright end-to-end suite for the 2026-07-06 Create-Flow overhaul.*
F1607 `Tlamatini/tests_e2e/test_ctrl_c_quits_visible.py` [188 lines] *VISIBLE end-to-end proof that Ctrl+C actually quits Tlamatini.*
F1608 `Tlamatini/tests_e2e/test_db_backup_set_visible.py` [416 lines] *Visible, dev-only Backup DB / Set DB round-trip regression test.*
F1609 `Tlamatini/tests_e2e/test_oob_browser_visible.py` [138 lines] *VISIBLE HEADED-BROWSER proof of the OOB_shift_reaper wiring, end-to-end.*
F1610 `Tlamatini/tests_e2e/test_oob_namu_unit.py` [190 lines] *OOB_shift_reaper + NAMU — deterministic unit proof of the FINAL design.*
F1611 `Tlamatini/tests_e2e/test_prompts_catalog_visual.py` [393 lines] *VISIBLE HEADED-BROWSER walk of the COMPLETE Catalog of Prompts (Angela, 2026-07-26).*
F1612 `Tlamatini/tests_e2e/test_self_healing_visual.py` [316 lines] *VISIBLE self-healing soak (Angela, 2026-07-06).*
F1613 `Tlamatini/tests_e2e/test_voice_visual.py` [322 lines] *Visible (HEADED) Playwright test for the VOICE feature (Config -> Voice).*
F1614 `Tlamatini/tests_e2e/Tlamatini-Whatsapper-Daily-Tests.py` [563 lines] *Tlamatini-Whatsapper-Daily-Tests.py – the definitive WhatsApp send tester.*
F1615 `Tlamatini/tests_e2e/torrenter_books.py` [151 lines] *Verify publisher PDFs through the authenticated source form, wrapped agent and live queue.*
F1616 `Tlamatini/tests_e2e/torrenter_console_windows.py` [90 lines] *Observe real console windows, including transient SHOW events, from a visible coordinator.*
F1617 `Tlamatini/tests_e2e/torrenter_live.py` [158 lines] *Visible development acceptance for the installed Jobs service and an enrolled test account.*
F1618 `Tlamatini/tests_e2e/torrenter_panel.py` [222 lines] *Run and inspect a real Torrenter flow using the visible Agentic Control Panel.*

repository root · guard tests — sheet 22

F0441 `test_author_banner.py` [98 lines] *Automated guard: Angela López Mendoza's authorship must be present EVERYWHERE. [4 tests]*
F0442 `test_check_private_data.py` [1.0k lines] *test_check_private_data.py – 100+ automated tests for the god-of-gods auditor. [143 tests]*
F0443 `test_perf_3x_visual.py` [479 lines] *VISUAL performance dashboard for the 3X-speed surgical plan.*
F0444 `test_private_data_guard.py` [287 lines] *Automated tests for the PRIVATE DATA GUARD mechanism. [9 tests]*

scripts/ · visible checks, release checks and tooling — sheet 22

F0338 `scripts/acp_editor_visible.py` [291 lines] *Exercise the real ACP editor in visible Chrome on port 8001.*
F0339 `scripts/agent_catalog_visible.py` [203 lines] *Check the live agent catalog and editor in visible Chrome on an isolated DB.*
F0343 `scripts/build_canvas_editor.mjs` [63 lines] *Run from a verified, visible foreground PowerShell -NoExit console.*
F0344 `scripts/build_setup_full_payload.py` [70 lines] *Build the current complete payload in a disposable, private source snapshot.*
F0345 `scripts/build_setup_validation.py` [58 lines] *Freeze the current setup clients and attach the newly built full payload.*
F0346 `scripts/canvas_editing_visible.py` [1.2k lines] *Real canvas editing and native saving, in verified visible Chrome only.*
F0347 `scripts/canvas_editor_entry.mjs` [248 lines] *The canvas's locally bundled text editor.*
F0348 `scripts/canvas_refresh_visible.py` [556 lines] *Canvas Refresh acceptance, on the real source page in visible Chrome only.*
F0349 `scripts/canvas_text_decoding_visible.py` [226 lines] *Open a real Markdown file in the source canvas, visibly and without changing it.*
F0350 `scripts/chat_composer_resize_checks.py` [264 lines] *Headed real-page layout checks.*
F0351 `scripts/chat_composer_resize_visible.py` [158 lines] *Composer geometry regression on the real source page and an isolated database.*
F0352 `scripts/chat_composer_visible_window.py` [83 lines] *Native Windows browser sizing: no viewport emulation or off-screen evidence.*
F0353 `scripts/chat_microphone_visible.py` [365 lines] *Visible real-template dictation checks with deterministic transport fixtures.*
F0354 `scripts/chat_viewport_visible_checks.py` [156 lines] *Native zoom and narrow-window checks, run in the visible source harness.*
F0355 `scripts/chat_views_frozen_visible.py` [254 lines] *Focused Views acceptance against a real frozen test instance on port 8878.*
F0356 `scripts/chat_views_visible.py` [480 lines] *Views acceptance on source/dev artifacts, never a rebuild or installed app.*
F0357 `scripts/check_js_parse.mjs` [68 lines] *Parse every owned JavaScript file and every shipped browser vendor file.*
F0358 `scripts/check_python_quality.py` [104 lines] *Lint and compile every owned Python source file from a visible console.*
F0359 `scripts/check_requirements_coverage.py` [283 lines] *Static repository-wide Python dependency coverage; imports no application code.*
F0360 `scripts/docked_avatar_resize_checks.py` [178 lines] *Docked portrait regression checks, called only by the headed source harness.*
F0361 `scripts/dossier_visual_walkthrough_visible.py` [231 lines] *VISIBLE walkthrough of the regenerated project dossier (PDF + PPTX).*
F0362 `scripts/drag_divider_look_visible.py` [467 lines] *VISIBLE proof that #drag-divider has the same look and feel as #vertical-drag-divider.*
F0363 `scripts/file_drop_router_cases.mjs` [455 lines] *file_drop_router_cases.mjs — behaviour tests for static/agent/js/file_drop_router.js*
F0364 `scripts/flow_canvas_mechanics_visible.py` [398 lines] *Shared editor checks, invoked only by the verified visible Chrome harness.*
F0365 `scripts/flow_file_associations_test.ps1` [120 lines] *Run in a verified visible foreground PowerShell -NoExit console.*
F0366 `scripts/flow_files_visible.py` [502 lines] *Visible real-browser flow-file lifecycle checks; launch via PowerShell -NoExit.*
F0367 `scripts/installed_panels_visible.py` [575 lines] *Visible acceptance checks against actual frozen or isolated source Tlamatini.*
F0368 `scripts/interaction_settings_visible.py` [354 lines] *Actual UI, editor and ConPTY acceptance.*
F0369 `scripts/interactor_runtime_server.py` [106 lines] *Test driver loaded by manage.py shell (source or the actual frozen EXE).*
F0370 `scripts/interactor_runtime_visible.py` [391 lines] *Focused source / existing frozen Interactor acceptance, never headless.*
F0371 `scripts/interactor_visible.py` [764 lines] *Visible Interactor acceptance.*
F0372 `scripts/markdown_canvas_visible.py` [573 lines] *Markdown canvas acceptance on the real source page, in visible Chrome only.*
F0373 `scripts/menu_browser_checks.py` [525 lines] *Real-template/browser state regressions.*
F0374 `scripts/menu_live_checks.py` [145 lines] *Live menu checks in the real application, through visible Chrome.*
F0375 `scripts/menu_reference_graph.mjs` [95 lines] *Run only in the verified visible test console.*
F0376 `scripts/panel_search_title_visible.py` [367 lines] *Real panel UI checks in visible Chrome and a verified foreground console.*
F0377 `scripts/playwright_real_visibility.cjs` [21 lines] *Opt-in preload for the visible Interactor harness only.*
F0378 `scripts/pptxer_visibility_gallery.py` [95 lines] *Create an offline, searchable gallery of actual PowerPoint test renders.*
F0379 `scripts/prompt_flow_commentary_visible.py` [175 lines] *Static commentary / unchanged user-input regression in real visible Chrome.*
F0380 `scripts/prompt_flow_connections_visible.py` [540 lines] *Verify ACP/Prompt Flow connector parity in VISIBLE Chrome on port 8001.*
F0381 `scripts/prompt_flow_extension_visible.py` [233 lines] *Exercise the real Prompt Flow Panel file format in visible Chrome on port 8001.*
F0382 `scripts/prompt_flow_output_resize_visible.py` [381 lines] *Run output splitter regression in real visible Chrome.*
F0383 `scripts/prompt_flow_selection_visible.py` [201 lines] *Exercise Prompt Flow Panel selection and status colors in visible Chrome on port 8001.*
F0384 `scripts/refresh_supporting_pdfs.py` [60 lines] *Regenerate the lifecycle note and the existing PDF renderer fixture visibly.*
F0385 `scripts/release_acp_editor_checks.py` [186 lines] *Real ACP graphical editing checks shared by source and frozen acceptance.*
F0386 `scripts/release_acp_runtime_checks.py` [79 lines] *Start, pause, resume and stop a real harmless ACP workflow through its UI.*
F0387 `scripts/release_agent_catalog_checks.py` [146 lines] *Inspect every actual ACP agent and round-trip its file through the visible UI.*
F0388 `scripts/release_auth_checks.py` [102 lines] *Normal UI authentication and disposable nonstaff user lifecycle checks.*
F0389 `scripts/release_commentary_checks.py` [401 lines] *Shared real-UI rich-commentary acceptance for source and frozen runs.*
F0390 `scripts/release_extended_checks.py` [295 lines] *Real, headed UI acceptance cases; never inject application state or responses.*
F0391 `scripts/release_local_agent_checks.py` [114 lines] *Execute local file agents through the product runtime in source or frozen mode.*
F0392 `scripts/release_mcp_checks.py` [81 lines] *Read-only client probes inside the actual source/frozen Django shell.*
F0393 `scripts/release_output_resize_checks.py` [230 lines] *Real playback and complete divider checks for source and frozen acceptance.*
F0394 `scripts/release_prompt_runtime_checks.py` [143 lines] *Real model, scheduling, branching and cancellation through Prompt Flow UI.*
F0395 `scripts/release_shell_file_checks.py` [73 lines] *Exercise the installed Windows file association using actual ShellExecute.*
F0396 `scripts/review_documentation_visuals.py` [84 lines] *Render every documentation PDF and produce numbered review sheets visibly.*
F0397 `scripts/run_menu_state_checks.py` [290 lines] *Deep menu suite: launch in a verified visible PowerShell -NoExit console.*
F0398 `scripts/setup_crash_agents_visible.py` [274 lines] *Real frozen-app crash campaign, from a verified visible -NoExit console.*
F0399 `scripts/setup_desktop_cycles.py` [138 lines] *Five real desktop lifecycle cases, started manually in a visible console.*
F0400 `scripts/setup_lifecycle_checks.py` [390 lines] *Focused real-filesystem/Windows checks; launch from a visible -NoExit console.*
F0401 `scripts/setup_release_cycles.py` [461 lines] *Visible, compiled setup validation.*
F0402 `scripts/setup_wizard_visible.py` [274 lines] *Visible source UI validation, NOT compiled release acceptance.*
F0403 `scripts/skill_inventory.py` [176 lines] *Generate the CURRENT skill inventory + version facts from Git and disk.*
F0404 `scripts/sync_assistant_skills.py` [143 lines] *Keep .gemini/skills/ in step with .claude/skills/.*
F0405 `scripts/terminator_visible_acp.py` [461 lines] *VISIBLE suite: the Terminals area AND the Terminater agent on the Agentic Control Panel.*
F0406 `scripts/terminator_visible_all.py` [47 lines] *VISIBLE all-in-one run: the chat page (with and without its canvas), the Agentic Control Panel (empty and with a loaded flow) and the Prompt Flow…*
F0407 `scripts/terminator_visible_battery.py` [733 lines] *The reusable VISIBLE check battery for the Terminals dock (any page that hosts it).*
F0408 `scripts/terminator_visible_chat.py` [228 lines] *VISIBLE suite: the Terminals area on the chat page (agent_page.html).*
F0409 `scripts/terminator_visible_common.py` [556 lines] *Shared harness for the VISIBLE Terminater suites (chat, ACP, Prompt Flow Panel).*
F0410 `scripts/terminator_visible_pfp.py` [402 lines] *VISIBLE suite: the Terminals area AND the Terminater step on the Prompt Flow Panel.*
F0411 `scripts/tlamatini_clean_install.ps1` [106 lines] *Destructive reset for an explicitly authorized CLEAN INSTALL test.*
F0412 `scripts/ui_sweep_resize_checks.py` [69 lines] *Behavior checks run only in the verified, headed ui_sweep_visible session.*
F0413 `scripts/ui_sweep_visible.py` [138 lines] *Interactive source UI audit, from a verified visible PowerShell -NoExit console.*
F0414 `scripts/update_flow_catalog.py` [100 lines] *Regenerate/check the standalone agent catalog and its documentation inventory.*
F0415 `scripts/usage_dashboard_browser_checks.py` [83 lines] *Visible real-page checks; deterministic data cases are explicitly labelled.*
F0416 `scripts/usage_dashboard_fixture_checks.py` [383 lines] *Explicit synthetic-data UI cases, supplemental to the real-source checks.*
F0417 `scripts/usage_dashboard_visible.py` [172 lines] *Real source app + isolated database, in a verified visible foreground console.*
F0418 `scripts/vendor_frontend.py` [95 lines] *Reproduce local UI dependencies from version/integrity-pinned npm archives.*
F0419 `scripts/vendor_pdfjs.py` [57 lines] *Reproduce the offline PDF canvas assets from the pinned upstream npm archive.*
F0420 `scripts/verify_latexer_styles.py` [246 lines] *Compile real LaTeX, audit content/geometry, render with Poppler, build an atlas.*
F0421 `scripts/verify_pdfer_styles.py` [151 lines] *Build, measure and render every PDFer style, plus a browsable gallery and atlas.*
F0422 `scripts/verify_pptxer_layout.py` [164 lines] *Reproducible PPTXer stress decks and a browsable PowerPoint render gallery.*
F0423 `scripts/verify_pptxer_styles.py` [133 lines] *Render the twelve new PPTXer styles with complete 256-character fields.*
F0424 `scripts/visual_fatal_dialog_visible.py` [191 lines] *Visible component integration checks using the real dialog, poller and styles.*

.claude/…/tlamatini-daily-chat-test/harness/ — sheet 22

F0090 `.claude/skills/tlamatini-daily-chat-test/harness/_peek.py` [27 lines] *Module that sets p, n, raw, txt using re, sys.*
F0091 `.claude/skills/tlamatini-daily-chat-test/harness/_stage.ps1` [75 lines] *Stage the 2x2 test desktop for Angela.*
F0092 `.claude/skills/tlamatini-daily-chat-test/harness/acp_backend_banner_test.py` [358 lines] *VISIBLE TEST — the "backend is down" banner in the ACP designer (ENGLISH)*
F0093 `.claude/skills/tlamatini-daily-chat-test/harness/acp_backend_banner_test.py.antes_shoter` [362 lines] *Pre-Shoter copy of acp_backend_banner_test.py kept as a comparison baseline.*
F0094 `.claude/skills/tlamatini-daily-chat-test/harness/acp_dialog_full_sweep.py` [377 lines] *ACP DIALOG STANDARDIZATION - FULL SWEEP, DRIVEN BY PLAYWRIGHTER*
F0095 `.claude/skills/tlamatini-daily-chat-test/harness/acp_dialog_theme_visible.py` [269 lines] *ACP RIGHT-CLICK DIALOGS - VISIBLE CHECK, DRIVEN BY PLAYWRIGHTER*
F0096 `.claude/skills/tlamatini-daily-chat-test/harness/ask_execs_regression.py` [676 lines] *ASK-EXECS REGRESSION SUITE — the Proceed/Deny permission scheme, end to end.*
F0097 `.claude/skills/tlamatini-daily-chat-test/harness/ask_execs_regression.py.antes_shoter` [679 lines] *Pre-Shoter copy of ask_execs_regression.py kept as a comparison baseline.*
F0098 `.claude/skills/tlamatini-daily-chat-test/harness/bulk_toggle_behaviour.py` [291 lines] *BEHAVIOURAL proof of the SPACE-bar bulk checkbox toggle, in a REAL browser.*
F0099 `.claude/skills/tlamatini-daily-chat-test/harness/bulk_toggle_visible.py` [199 lines] *VISIBLE proof of the SPACE-bar bulk checkbox toggle.*
F0100 `.claude/skills/tlamatini-daily-chat-test/harness/cancel_regression.py` [571 lines] *HARD-CANCEL REGRESSION TEST — visible, headed, on Angela's real desktop.*
F0101 `.claude/skills/tlamatini-daily-chat-test/harness/cancel_regression.py.antes_shoter` [575 lines] *Pre-Shoter copy of cancel_regression.py kept as a comparison baseline.*
F0102 `.claude/skills/tlamatini-daily-chat-test/harness/compact_mode_visible.py` [483 lines] *VISIBLE proof: COMPACT MODE - one-shot questions work on the SMALL model the installed Tlamatini is configured with (Angela, 2026-10-01).*
F0103 `.claude/skills/tlamatini-daily-chat-test/harness/compact_pingpong_visible.py` [342 lines] *TWO CHAT PAGES, TWO MODELS - THE COMPACT-MODE VERDICT MUST NOT PING-PONG (Angela Lopez Mendoza, 2026-10-08)*
F0104 `.claude/skills/tlamatini-daily-chat-test/harness/compact_switch_visible.py` [691 lines] *VISIBLE proof: the COMPACT MODE checkbox (Angela, 2026-10-02).*
F0105 `.claude/skills/tlamatini-daily-chat-test/harness/config.py` [132 lines] *Central configuration + DOM contract for the Tlamatini daily chat test.*
F0106 `.claude/skills/tlamatini-daily-chat-test/harness/context_gauge_100.py` [500 lines] *100+ REAL prompts in a VISIBLE Chrome, watching the context gauge move.*
F0107 `.claude/skills/tlamatini-daily-chat-test/harness/context_gauge_real_lab.py` [508 lines] *THE REAL CONTEXT GAUGE - VISIBLE LABORATORY (Angela, 2026-09-28)*
F0108 `.claude/skills/tlamatini-daily-chat-test/harness/context_gauge_visible.py` [288 lines] *VISIBLE proof that the context gauge moves on a REAL request.*
F0109 `.claude/skills/tlamatini-daily-chat-test/harness/context_restore_spinner_visible.py` [726 lines] *context_restore_spinner_visible.py - VISIBLE regression runner (2026-09-22)*
F0110 `.claude/skills/tlamatini-daily-chat-test/harness/cyber_watchdog_test.py` [463 lines] *Cybersecurity / command-watchdog stress test for Tlamatini (VISIBLE Chrome).*
F0111 `.claude/skills/tlamatini-daily-chat-test/harness/deepseek_remote_chat_visible.py` [616 lines] *VISIBLE proof: the whole chat works against a REMOTE Ollama serving deepseek-coder.*
F0112 `.claude/skills/tlamatini-daily-chat-test/harness/dialog_policy_visible.py` [493 lines] *ESCAPE CLOSES EVERY DIALOG - VISIBLE REGRESSION, DRIVEN BY PLAYWRIGHTER*
F0113 `.claude/skills/tlamatini-daily-chat-test/harness/dialog_theme_visible.py` [256 lines] *DIALOG STANDARDIZATION - VISIBLE REGRESSION, DRIVEN BY PLAYWRIGHTER*
F0114 `.claude/skills/tlamatini-daily-chat-test/harness/discoverer_1000.py` [27 lines] *Tlamatini — 1000 Discoverer tests through the REAL chat GUI, with full-screen photographic evidence (entire desktop incl. the taskbar CLOCK) for…*
F0115 `.claude/skills/tlamatini-daily-chat-test/harness/discoverer_1000.py.antes_shoter` [454 lines] *Pre-Shoter copy of discoverer_1000.py kept as a comparison baseline.*
F0116 `.claude/skills/tlamatini-daily-chat-test/harness/drop_message_visible.py` [670 lines] *THE DROP BUTTON - VISIBLE END-TO-END TEST (Angela Lopez Mendoza, 2026-09-30)*
F0117 `.claude/skills/tlamatini-daily-chat-test/harness/file_drop_visible.py` [615 lines] *FILE DROP ROUTER - VISIBLE END-TO-END TEST (Angela, 2026-10-14)*
F0118 `.claude/skills/tlamatini-daily-chat-test/harness/flowcreator_wizard_e2e.py` [374 lines] *Tlamatini — VISIBLE (headed Chrome) end-to-end test of the NEW catalog prompt "FLOWCREATOR END-TO-END WIZARD" (prompt #108, Agents & Flows).*
F0119 `.claude/skills/tlamatini-daily-chat-test/harness/full_mode_visible.py` [252 lines] *FULL MODE on a big cloud model - VISIBLE E2E (Angela Lopez Mendoza, 2026-10-01).*
F0120 `.claude/skills/tlamatini-daily-chat-test/harness/googler_dork_hunt.py` [324 lines] *GOOGLER DORK HUNT — a named, re-runnable, VISIBLE proof of the dork builder.*
F0121 `.claude/skills/tlamatini-daily-chat-test/harness/grepper_lines_visible.py` [235 lines] *VISIBLE end-to-end proof that the modified Grepper did NOT break Tlamatini.*
F0122 `.claude/skills/tlamatini-daily-chat-test/harness/grepper_login_probe.py` [62 lines] *VISIBLE login probe - prints exactly where the chat page is lost.*
F0123 `.claude/skills/tlamatini-daily-chat-test/harness/latexer_openmp_e2e.py` [446 lines] *VISIBLE end-to-end replay of Angela's OpenMP LaTeX request.*
F0124 `.claude/skills/tlamatini-daily-chat-test/harness/latexer_stepbystep_5.py` [861 lines] *Tlamatini - VISIBLE (headed Chrome) end-to-end test of FIVE escalating LaTeXer STEP-BY-STEP wizards, with a TRUTH ORACLE over the Exec Report.*
F0125 `.claude/skills/tlamatini-daily-chat-test/harness/mcp_playwright_suite.py` [218 lines] *Tlamatini — 10-MCP Playwright suite (visible Chrome).*
F0126 `.claude/skills/tlamatini-daily-chat-test/harness/model_brain_visible.py` [557 lines] *MODEL BRAIN - VISIBLE end-to-end test (Angela Lopez Mendoza, 2026-10-08).*
F0127 `.claude/skills/tlamatini-daily-chat-test/harness/monitor.py` [178 lines] *LIVE TEST MONITOR - a window that REFRESHES ITSELF while the test runs.*
F0128 `.claude/skills/tlamatini-daily-chat-test/harness/ollama_remote_catalog_visible.py` [505 lines] *VISIBLE proof: Config > Models asks the CONFIGURED (remote) Ollama.*
F0129 `.claude/skills/tlamatini-daily-chat-test/harness/pdf_context_visible.py` [424 lines] *PDF AS CONTEXT - VISIBLE TEST (Angela Lopez Mendoza, 2026-10-08)*
F0130 `.claude/skills/tlamatini-daily-chat-test/harness/pdfer_nuance_visible.py` [529 lines] *PDFer NUANCE / LAYOUT — THE VISIBLE TEST*
F0131 `.claude/skills/tlamatini-daily-chat-test/harness/playwrighter_run.py` [164 lines] *PLAYWRIGHTER LAUNCHER for the visible harnesses.*
F0132 `.claude/skills/tlamatini-daily-chat-test/harness/pptxer_visible.py` [781 lines] *PPTXer — THE VISIBLE TEST*
F0133 `.claude/skills/tlamatini-daily-chat-test/harness/preflight.py` [334 lines] *Make a visible test INVULNERABLE to peripheral failure.*
F0134 `.claude/skills/tlamatini-daily-chat-test/harness/qualify.py` [188 lines] *Answer qualification for the Tlamatini daily chat test.*
F0135 `.claude/skills/tlamatini-daily-chat-test/harness/questions.py` [403 lines] *Question bank generator for the Tlamatini daily chat test.*
F0136 `.claude/skills/tlamatini-daily-chat-test/harness/README.md` [118 lines] *Tlamatini Daily Chat Test — harness: A Playwright harness that drives real Chrome, logs into Tlamatini, and asks up to 1000 curated, safe-to-execute questions…*
F0137 `.claude/skills/tlamatini-daily-chat-test/harness/requirements.txt` [7 lines] *Python dependency list: 2 pinned requirements.*
F0138 `.claude/skills/tlamatini-daily-chat-test/harness/run_test.py` [726 lines] *Tlamatini daily chat test – Playwright harness (visible Chrome).*
F0139 `.claude/skills/tlamatini-daily-chat-test/harness/self_modify_visible.py` [481 lines] *The SELF-MODIFY box - VISIBLE end-to-end test (Angela Lopez Mendoza, 2026-10-03).*
F0140 `.claude/skills/tlamatini-daily-chat-test/harness/shoter_shot.py` [100 lines] *TEST SCREENSHOTS ARE TAKEN BY SHOTER, TLAMATINI'S OWN AGENT*
F0141 `.claude/skills/tlamatini-daily-chat-test/harness/step_by_step_catalog.py` [381 lines] *Tlamatini — VISIBLE (headed Chrome) end-to-end test of the Catalog-of-Prompts STEP-BY-STEP wizards (the section openers added by migrations 0181 +…*
F0142 `.claude/skills/tlamatini-daily-chat-test/harness/step_by_step_catalog.py.antes_shoter` [381 lines] *Pre-Shoter copy of step_by_step_catalog.py kept as a comparison baseline.*
F0143 `.claude/skills/tlamatini-daily-chat-test/harness/stop_sequence_visible.py` [582 lines] *TOO MANY STOP SEQUENCES - VISIBLE end-to-end test (Angela Lopez Mendoza, 2026-10-03).*
F0144 `.claude/skills/tlamatini-daily-chat-test/harness/talk_test.py` [397 lines] *Tlamatini TALK test – Playwright harness (visible Chrome).*
F0145 `.claude/skills/tlamatini-daily-chat-test/harness/uninstaller_visible.py` [466 lines] *uninstaller_visible.py — VISIBLE end-to-end test of the Tlamatini Uninstaller.*
F0146 `.claude/skills/tlamatini-daily-chat-test/harness/video_link_visible.py` [369 lines] *VIDEO LINK TRANSCRIPTION - VISIBLE TEST (Angela Lopez Mendoza, 2026-10-11)*
F0147 `.claude/skills/tlamatini-daily-chat-test/harness/voice_commands_visible.py` [420 lines] *VISIBLE proof of the VOICE COMMANDS catalog section (Angela, 2026-09-12).*
F0148 `.claude/skills/tlamatini-daily-chat-test/harness/window-test-screen.py` [1.5k lines] *WINDOW TEST SCREEN - a graphical instrument panel Claude drives in real time.*
F0149 `.claude/skills/tlamatini-daily-chat-test/harness/windower_focus.py` [164 lines] *THE WINDOW IS BROUGHT FORWARD BY WINDOWER, TLAMATINI'S OWN AGENT*
F0150 `.claude/skills/tlamatini-daily-chat-test/harness/wrapped_questions.py` [323 lines] *One functional question per WRAPPED chat-agent in Tlamatini.*

.gemini/…/tlamatini-daily-chat-test/harness/ — sheet 22

F0164 `.gemini/skills/tlamatini-daily-chat-test/harness/acp_backend_banner_test.py` [358 lines] *VISIBLE TEST — the "backend is down" banner in the ACP designer (ENGLISH)*
F0165 `.gemini/skills/tlamatini-daily-chat-test/harness/acp_backend_banner_test.py.antes_shoter` [362 lines] *Pre-Shoter copy of acp_backend_banner_test.py kept as a comparison baseline.*
F0166 `.gemini/skills/tlamatini-daily-chat-test/harness/acp_dialog_full_sweep.py` [377 lines] *ACP DIALOG STANDARDIZATION - FULL SWEEP, DRIVEN BY PLAYWRIGHTER*
F0167 `.gemini/skills/tlamatini-daily-chat-test/harness/acp_dialog_theme_visible.py` [269 lines] *ACP RIGHT-CLICK DIALOGS - VISIBLE CHECK, DRIVEN BY PLAYWRIGHTER*
F0168 `.gemini/skills/tlamatini-daily-chat-test/harness/ask_execs_regression.py` [676 lines] *ASK-EXECS REGRESSION SUITE — the Proceed/Deny permission scheme, end to end.*
F0169 `.gemini/skills/tlamatini-daily-chat-test/harness/ask_execs_regression.py.antes_shoter` [679 lines] *Pre-Shoter copy of ask_execs_regression.py kept as a comparison baseline.*
F0170 `.gemini/skills/tlamatini-daily-chat-test/harness/bulk_toggle_behaviour.py` [291 lines] *BEHAVIOURAL proof of the SPACE-bar bulk checkbox toggle, in a REAL browser.*
F0171 `.gemini/skills/tlamatini-daily-chat-test/harness/bulk_toggle_visible.py` [199 lines] *VISIBLE proof of the SPACE-bar bulk checkbox toggle.*
F0172 `.gemini/skills/tlamatini-daily-chat-test/harness/cancel_regression.py` [571 lines] *HARD-CANCEL REGRESSION TEST — visible, headed, on Angela's real desktop.*
F0173 `.gemini/skills/tlamatini-daily-chat-test/harness/cancel_regression.py.antes_shoter` [575 lines] *Pre-Shoter copy of cancel_regression.py kept as a comparison baseline.*
F0174 `.gemini/skills/tlamatini-daily-chat-test/harness/config.py` [132 lines] *Central configuration + DOM contract for the Tlamatini daily chat test.*
F0175 `.gemini/skills/tlamatini-daily-chat-test/harness/cyber_watchdog_test.py` [463 lines] *Cybersecurity / command-watchdog stress test for Tlamatini (VISIBLE Chrome).*
F0176 `.gemini/skills/tlamatini-daily-chat-test/harness/dialog_policy_visible.py` [493 lines] *ESCAPE CLOSES EVERY DIALOG - VISIBLE REGRESSION, DRIVEN BY PLAYWRIGHTER*
F0177 `.gemini/skills/tlamatini-daily-chat-test/harness/dialog_theme_visible.py` [256 lines] *DIALOG STANDARDIZATION - VISIBLE REGRESSION, DRIVEN BY PLAYWRIGHTER*
F0178 `.gemini/skills/tlamatini-daily-chat-test/harness/discoverer_1000.py` [27 lines] *Tlamatini — 1000 Discoverer tests through the REAL chat GUI, with full-screen photographic evidence (entire desktop incl. the taskbar CLOCK) for…*
F0179 `.gemini/skills/tlamatini-daily-chat-test/harness/discoverer_1000.py.antes_shoter` [454 lines] *Pre-Shoter copy of discoverer_1000.py kept as a comparison baseline.*
F0180 `.gemini/skills/tlamatini-daily-chat-test/harness/flowcreator_wizard_e2e.py` [374 lines] *Tlamatini — VISIBLE (headed Chrome) end-to-end test of the NEW catalog prompt "FLOWCREATOR END-TO-END WIZARD" (prompt #108, Agents & Flows).*
F0181 `.gemini/skills/tlamatini-daily-chat-test/harness/googler_dork_hunt.py` [324 lines] *GOOGLER DORK HUNT — a named, re-runnable, VISIBLE proof of the dork builder.*
F0182 `.gemini/skills/tlamatini-daily-chat-test/harness/latexer_openmp_e2e.py` [446 lines] *VISIBLE end-to-end replay of Angela's OpenMP LaTeX request.*
F0183 `.gemini/skills/tlamatini-daily-chat-test/harness/latexer_stepbystep_5.py` [861 lines] *Tlamatini - VISIBLE (headed Chrome) end-to-end test of FIVE escalating LaTeXer STEP-BY-STEP wizards, with a TRUTH ORACLE over the Exec Report.*
F0184 `.gemini/skills/tlamatini-daily-chat-test/harness/mcp_playwright_suite.py` [218 lines] *Tlamatini — 10-MCP Playwright suite (visible Chrome).*
F0185 `.gemini/skills/tlamatini-daily-chat-test/harness/playwrighter_run.py` [164 lines] *PLAYWRIGHTER LAUNCHER for the visible harnesses.*

F0186 `.gemini/skills/tlamatini-daily-chat-test/harness/qualify.py` [188 lines] *Answer qualification for the Tlamatini daily chat test.*
F0187 `.gemini/skills/tlamatini-daily-chat-test/harness/questions.py` [403 lines] *Question bank generator for the Tlamatini daily chat test.*
F0188 `.gemini/skills/tlamatini-daily-chat-test/harness/README.md` [118 lines] *Tlamatini Daily Chat Test — harness: A Playwright harness that drives real Chrome, logs into Tlamatini, and asks up to 1000 curated, safe-to-execute questions…*
F0189 `.gemini/skills/tlamatini-daily-chat-test/harness/requirements.txt` [7 lines] *Python dependency list: 2 pinned requirements.*
F0190 `.gemini/skills/tlamatini-daily-chat-test/harness/run_test.py` [726 lines] *Tlamatini daily chat test – Playwright harness (visible Chrome).*
F0191 `.gemini/skills/tlamatini-daily-chat-test/harness/shoter_shot.py` [100 lines] *TEST SCREENSHOTS ARE TAKEN BY SHOTER, TLAMATINI'S OWN AGENT*
F0192 `.gemini/skills/tlamatini-daily-chat-test/harness/step_by_step_catalog.py` [381 lines] *Tlamatini — VISIBLE (headed Chrome) end-to-end test of the Catalog-of-Prompts STEP-BY-STEP wizards (the section openers added by migrations 0181 +…*
F0193 `.gemini/skills/tlamatini-daily-chat-test/harness/step_by_step_catalog.py.antes_shoter` [381 lines] *Pre-Shoter copy of step_by_step_catalog.py kept as a comparison baseline.*
F0194 `.gemini/skills/tlamatini-daily-chat-test/harness/talk_test.py` [397 lines] *Tlamatini TALK test – Playwright harness (visible Chrome).*
F0195 `.gemini/skills/tlamatini-daily-chat-test/harness/wrapped_questions.py` [323 lines] *One functional question per WRAPPED chat-agent in Tlamatini.*

Atlas VIII — Build, release, install, security and repository root

build and release tooling — sheet 23

F0201 `.gitattributes` [14 lines] *Git attributes: line-ending and binary handling.*
F0203 `.gitignore` [462 lines] *Git ignore rules: Temp/, Templates/, databases, secrets (data.keys), build output, generated _version.py.*
F0211 `audit_dependencies.py` [185 lines] *Tlamatini dependency audit — run it on ANY interpreter, it changes nothing.*
F0216 `build.py` [2.4k lines] *build.py — Tlamatini Build Script*
F0217 `build_all.cmd` [101 lines] *build_all.cmd — Build Tlamatini end-to-end.*
F0218 `build_complete_private_release.py` [546 lines] *build_complete_private_release.py – KEYED (private) release builder.*
F0219 `build_complete_public_release.py` [956 lines] *build_complete_public_release.py – PUBLIC release builder (scrubbed + verified).*
F0221 `build_installer.py` [564 lines] *build_installer.py — Tlamatini Installer Build Script*
F0222 `build_runtime_assets.py` [524 lines] *Fail-closed runtime asset inventory shared by the application/installer builds.*
F0223 `build_uninstaller.py` [371 lines] *build_uninstaller.py — Tlamatini Uninstaller Build Script*
F0224 `check_private_data.py` [721 lines] *check_private_data.py – god-of-gods private-data leak auditor.*
F0229 `copy_source_assets.py` [984 lines] *copy_source_assets.py — generate the TlamatiniSourceCode self-modify snapshot.*
F0305 `eslint.config.mjs` [537 lines] *JavaScript module.*
F0311 `git_deny_go.py` [154 lines] *git_deny_go.py — autonomously guarantee the Go toolchain NEVER appears in git.*
F0321 `package.json` [52 lines] *Node tooling manifest (tlamatini): 24 dependencies, scripts build:canvas-editor, lint, lint:fix, lint:python.*
F0323 `pnpm-lock.yaml` [1.3k lines] *pnpm lockfile pinning the JavaScript lint tooling.*
F0325 `private_targets.example.json` [43 lines] *JSON document with keys: _README, names, emails, phones, handles.*
F0328 `pyinstaller_hooks/hook-numpy.py` [158 lines] *Tlamatini custom PyInstaller hook for numpy.*
F0329 `pyinstaller_hooks/hook-pymupdf.py` [14 lines] *Bundle native MuPDF libraries used by PDF context in the frozen web process.*
F0330 `pyinstaller_hooks/hook-torch.py` [30 lines] *Prevent PyInstaller from collecting Torch assets for the frozen web app.*
F0332 `regen_secrets.py` [567 lines] *regen_secrets.py — toggle Tlamatini config files between two modes*
F0336 `requirements-esphome.txt` [7 lines] *Python dependency list: 1 pinned requirements (private ESPHome runtime).*
F0337 `requirements.txt` [304 lines] *Python dependency list: 264 pinned requirements.*
F1632 `uncommit-keys.bat` [243 lines] *uncommit-keys.bat - Angela's secret-leak seatbelt for the Tlamatini repo.*
F1638 `versioning.py` [233 lines] *Top-level versioning helper for Tlamatini's build scripts.*

install, update and launch — sheet 23

F0210 `apply_update.ps1` [389 lines] *Tlamatini Auto-Updater Runs OUTSIDE the install directory (a copy is placed under %LOCALAPPDATA%\Tlamatini\updater by self_update.py) so it can…*
F0214 `backup_db.py` [206 lines] *Daily backup of Tlamatini's database.*
F0230 `CreateShortcut.json` [3 lines] *JSON document with keys: InstallDir.*
F0231 `CreateShortcut.ps1` [264 lines] *Shortcut Creator for Tlamatini Reads the installation directory from CreateShortcut.json*
F0308 `flow_file_associations.ps1` [181 lines] *Per-user registration. Never modify UserChoice, its hash, or other applications.*
F0309 `freeingport8000.ps1` [65 lines] *PowerShell script.*
F0312 `install.py` [880 lines] *Defines resolve_version(), FancyInstaller.*
F0318 `ollama_credits.py` [387 lines] *ollama_credits.py - Zero-dependency Ollama Cloud credits/usage checker.*
F0324 `preserved_user_state.json` [28 lines] *JSON document with keys: _README, preserve, _notes.*
F0333 `register_flw.ps1` [15 lines] *Run visibly in PowerShell.*
F0334 `register_fpmt.ps1` [12 lines] *Run visibly in PowerShell.*
F0335 `RemoveShortcut.ps1` [59 lines] *Shortcut Remover for Tlamatini Removes the desktop and local shortcuts created by CreateShortcut.ps1 and forces Windows Explorer to refresh so the…*
F0436 `setup_logging.py` [100 lines] *Private local setup transcripts, including for windowed executables.*
F0437 `setup_recovery.py` [587 lines] *Bounded, installation-scoped recovery for the standalone setup clients.*
F0438 `setup_windows.py` [271 lines] *Small, explicitly scoped Windows setup primitives; no application imports.*
F0439 `setup_wizard.py` [529 lines] *Shared, standalone presentation for the Windows setup tools.*
F0470 `Tlamatini.ps1` [69 lines] *Tlamatini PowerShell Script Executes the Django development server with –noreload flag and automatically opens the browser on the configured web…*
F1633 `uninstall.py` [1.6k lines] *uninstall.py — Tlamatini Uninstaller*
F1634 `uninstall_processes.py` [147 lines] *Installation-scoped process cleanup for the standalone Windows uninstaller.*
F1635 `unregister_flw.ps1` [15 lines] *Run visibly in PowerShell.*
F1636 `unregister_fpmt.ps1` [12 lines] *Run visibly in PowerShell.*

root stdio MCP server — sheet 23

F0204 `.mcp.json` [8 lines] *JSON document with keys: mcpServers.*
F1627 `tlamatini_acpx.py` [986 lines] *Self-contained ACPX runtime for the Tlamatini MCP server.*
F1631 `tlamatini_mcp_server.py` [770 lines] *Tlamatini Agents — MCP server.*

security/ · security arsenal — sheet 23

F0425 `security/automated_tests_of_security_assets.py` [128 lines] *Non-admin regression checks, run only in a confirmed visible console.*
F0426 `security/enable_tlamatini_v2.bat` [734 lines] *Generated from tlamatini_whitelist_v2.ps1 + windows_access_helpers.ps1.*
F0427 `security/README.md` [139 lines] *Tlamatini Windows access and Blue-hat toolkit: Double-click enable_tlamatini_v2.bat when you want to apply Windows access settings, approve its UAC prompt…*
F0428 `security/run_defender.bat` [20 lines] *Default is report-only: no process kills and no firewall block additions.*
F0429 `security/sync_enable_launcher.py` [65 lines] *Regenerate the standalone enable launcher from its reviewed PowerShell sources.*
F0430 `security/test_security_syntax.ps1` [13 lines] *Parser only: no enablement or defender workloads are executed.*
F0431 `security/test_windows_access.ps1` [119 lines] *Non-admin behavior tests: every Windows-policy mutation below is a mock.*
F0432 `security/tlamatini_defender.ps1` [718 lines] *TLAMATINI DEFENDER - REPORT-ONLY BY DEFAULT, OPTIONAL RESPONSE (v2.2) Purpose: Real-time intrusion detection and automated defensive response.*
F0433 `security/tlamatini_whitelist_v2.ps1` [567 lines] *TLAMATINI SECURITY WHITELIST SCRIPT v2.3 - VERIFIED APPLICATION ACCESS Purpose: Adds Tlamatini to Windows security exclusions AND grants it…*
F0434 `security/verify_agent_execution.ps1` [132 lines] *Check real non-admin execution without running model jobs or applying policy.*
F0435 `security/windows_access_helpers.ps1` [128 lines] *Pure helper definitions.*

CI and scanning policy — sheet 23

F0202 `.github/workflows/name-guard.yml` [99 lines] *YAML document: keys name, on, permissions, concurrency, jobs.*
F0205 `.scanning/finding-policy.json` [94 lines] *JSON document with keys: bandit, semgrep.*

demo flows — sheet 23

F0316 `MCPMemoriesFlowCreation.flw` [434 lines] *Agentic flow (.flw) for the Agentic Control Panel with 17 nodes.*

repository root · media, documents and license — sheet 23

F0220 `build_env.py` [323 lines] *The SEALED build Python: what you install on this machine can never reach a release.*
F0228 `ConfigureModels.png` [146KB] *Image (PNG): ConfigureModels.*
F0315 `LICENSE` [21 lines] *MIT license of Tlamatini.*
F0319 `OllamaPricing.png` [178KB] *Image (PNG): OllamaPricing.*
F0320 `OOB_shift_reaper-and-NAMU-Design.pdf` [46KB] *PDF document: OOB shift reaper-and-NAMU-Design.*
F0326 `PromptDesigner.jpg` [106KB] *Image (JPG): PromptDesigner.*
F0327 `pyinstaller_guard.py` [211 lines] *Run PyInstaller so a Tlamatini release build cannot die on the build machine's clutter.*
F0468 `Tlamatini.ico` [38KB] *Tlamatini application icon.*
F0469 `Tlamatini.jpg` [15KB] *Image (JPG): Tlamatini.*
F1628 `tlamatini_app_summary.pdf` [2.8MB] *PDF document: tlamatini app summary.*
F1629 `Tlamatini_eXtended_Artificial_Intelligence_Humanly_Tempered.pptx` [5.0MB] *Project presentation deck generated by the dossier generator.*

Atlas IX — Knowledge and governance

repository root · Markdown and text documents — sheet 24

F0206 `_acpx_pipeline_demo_report.md` [65 lines] *ACPX Pipeline Demo Report: - Session ID: c697c78913ea4ab4b308211d9649d8bd - Agent: gemini (Google Gemini CLI v0.37.2) - Transport: tui-repl - Working…*
F0207 `ACPX.md` [910 lines] *ACPX & Skills in Tlamatini — A complete guide for dummies: Torrenter adds account-owned downloads to chat and the Agentic Control Panel.*
F0208 `AGENTS.md` [493 lines] *Tlamatini instructions for Codex and every coding agent: The combined builder predicts both editions' actual emitted version before building, pins one…*
F0209 `agents_descriptions.md` [265 lines] *Tlamatini — Agent Descriptions: Torrenter adds account-owned downloads to chat and the Agentic Control Panel.*
F0213 `AvatarImprovementByClaude.md` [416 lines] *The Avatar — from a 150 ms metronome to a talking head: Date: 2026-09-11 · Baseline: v1.51.5 · Author of Tlamatini: Angela López Mendoza Result: 74 / 74…*
F0215 `BookOfTlamatini.md` [4.7k lines] *Tlamatini: The combined builder predicts both editions' actual emitted version before building, pins one matching number, runs Private/Public pairs in…*
F0225 `Claude-Desktop-KALI-MCP-Session.md` [694 lines] *Claude Desktop ↔ Kali Linux MCP — Complete Setup Guide: Who this document is for: Anyone (human or Claude AI) setting up or maintaining the bridge between…*
F0226 `CLAUDE.md` [1.4k lines] *PRIVATE DATA GUARD — ABSOLUTE, NON-NEGOTIABLE, READ FIRST : The combined builder predicts both editions' actual emitted version before building, pins one…*
F0227 `Codex2ClaudeRecommendations.md` [624 lines] *Codex to Claude: Tlamatini skill audit and recommendations: Prepared by Codex on 2026-09-23 for Claude and Angela.*
F0232 `DesignOfIncludingMemoryMCPS.txt` [1.2k lines] *Tlamatini (2026/08/15 16:04:42.921)*
F0233 `Discoverier-new-agent.md` [277 lines] *Discoverer — New Agent Design (ProjectDiscovery suite bridge): Discoverer is a new Tlamatini workflow agent that runs the ProjectDiscovery security tool suite…*
F0306 `example_of_prompt_to_make_de-compresser_agent.txt` [42 lines] *Please read create_new_agent.md first and either analize the code of all of the agents in Tlamatini/agent/agents/* in order to you to understand how…*
F0307 `FirstFinalPlanToSpeedUp.md` [774 lines] *First Final Plan To Speed Up Tlamatini: Objective: implement one low-risk speedup batch that improves runtime speed without reducing Tlamatini's virtues…*
F0310 `GEMINI.md` [1.2k lines] *PRIVATE DATA GUARD — ABSOLUTE, NON-NEGOTIABLE, READ FIRST : The combined builder predicts both editions' actual emitted version before building, pins one…*
F0313 `KIMI.md` [1.4k lines] *PRIVATE DATA GUARD — ABSOLUTE, NON-NEGOTIABLE, READ FIRST : The combined builder predicts both editions' actual emitted version before building, pins one…*
F0314 `latest-folding-context-session.md` [4 lines] *10247944-d2ab-420e-b58c-1a7af83442e9*
F0317 `MessagingConnectorAssistant_Design.md` [134 lines] *MessagingConnectorAssistant — Design: Goal (Angela's): the user tells Tlamatini their phone number and *nothing else*; Tlamatini does every step to…*
F0322 `PIVOT_CHANGES.md` [569 lines] *Tlamatini — Change Pivot / Rollback Log: The combined builder predicts both editions' actual emitted version before building, pins one matching number, runs…*
F0331 `README.md` [365 lines] *v1.92.4 — Matching builds, explicit release identity and readable Markdown*
F0440 `test-reloaded-session.md` [4 lines] *10247944-d2ab-420e-b58c-1a7af83442e9*
F0465 `TestsVisiblesAndVisibleExecutionFromClaude2Codex.md` [230 lines] *Claude → Codex: EVERY TEST AND EVERY EXECUTION MUST BE VISIBLE: From: Claude, 2026-09-23 To: Codex, and every model that works on Tlamatini after it this to…*
F0467 `Tlamatini-Kali-Setup.md` [413 lines] *Tlamatini ↔ Kali Linux — Zero-Client Setup (Kalier): Who this is for: Anyone (human or AI) who wants Tlamatini to drive Kali Linux penetration-testing tools…*
F1630 `TLAMATINI_MCP.md` [292 lines] *Tlamatini Agents — MCP connector: Torrenter adds account-owned downloads to chat and the Agentic Control Panel.*
F1637 `VERSIONING.md` [470 lines] *Tlamatini — Versioning: When Angela specifies X.Y.Z, assume X.Y.Z is already released and use it everywhere as the latest released version immediately.*

docs/ · feature guides — sheet 24

F0234 `docs/agent-coverage.md` [116 lines] *Installed agent coverage: Generated by python scripts/update_flow_catalog.py; verify with –check.*
F0235 `docs/assets/readme-hero.svg` [114 lines] *SVG graphic: readme-hero.*
F0236 `docs/build-root-assets-verification.md` [162 lines] *Required runtime files: packaging repair and verification: Torrenter adds account-owned downloads to chat and the Agentic Control Panel.*
F0237 `docs/canvas-and-interactor-quick-reference.txt` [62 lines] *TLAMATINI - CANVAS EDITOR AND INTERACTOR QUICK REFERENCE*
F0238 `docs/canvas-editing.md` [183 lines] *Editing and saving canvas files: Introduced in v1.89.0 (2026-10-21), carried by v1.91.0 with configurable editor behavior.*
F0239 `docs/canvas-refresh.md` [110 lines] *Refresh the open canvas file: Open chooses a document.*
F0269 `docs/chat-microphone-design.md` [442 lines] *Direct voice prompts: 1.*
F0280 `docs/combined-release-builder.md` [115 lines] *Combined English and Spanish releases: The combined builder predicts both editions' actual emitted version before building, pins one matching number, runs…*
F0281 `docs/companion-app-discovery.md` [123 lines] *Tlamatini — Companion-App Discovery Contract: A stable lookup surface that lets XAIHT companion apps (e.g.*
F0282 `docs/dependency-coverage-audit.md` [101 lines] *Python dependency coverage and reproducible installation — 2026-10-18: The main requirements.txt describes the complete Windows CPython 3.12 environment used…*
F0283 `docs/desktop-input-and-flow-contracts.md` [127 lines] *Desktop input and flow contracts: Implementation update, 2026-09-15.*
F0284 `docs/examples/prompting-kickoff.fpmt` [26 lines] *Example Prompt Flow Panel diagram (JSON format tlamatini-prompting-flow).*
F0285 `docs/external_mcp_bulletproof_architecture.md` [141 lines] *External MCPs — the bulletproof architecture: The design contract for Tlamatini's universal MCP _client_: the layer that lets her use the tools of any…*
F0286 `docs/GUI-Manager-design.md` [344 lines] *Mouser: read physical cursor and monitor layout without moving anything.*
F0287 `docs/interaction-settings.md` [160 lines] *Live Editor and Terminater settings: Since v1.92.0, the maximum is 250 WPM for each control.*
F0288 `docs/interactor.md` [245 lines] *Interactor: teaching in the live canvas editor: Interactor is a module, not a workflow agent.*
F0289 `docs/markdown-canvas.md` [86 lines] *Markdown in the chat canvas: A readable Markdown file is not rejected merely for containing an isolated control character, such as a bell or backspace…*
F0290 `docs/model-configuration-verification.md` [44 lines] *Runtime and model configuration verification — 2026-09-20: Installed application: C:\Tlamatini\Tlamatini.exe, version 1.63.0.*
F0291 `docs/model_configuration.md` [338 lines] *Model configuration: Config → Models manages 38 model, engine and voice settings for core services and 21 model-backed agents.*
F0292 `docs/oob-namu-lifecycle.md` [85 lines] *OOB_shift_reaper and NAMU — Tlamatini 1.92.4: Reviewed against source a771d27c, 2026-10-23; tag v1.91.0 resolves to the same commit.*
F0293 `docs/pdf-canvas.md` [260 lines] *PDF canvas and document context: Views → Canvas keeps this document and its viewer alive while hiding the pane.*
F0294 `docs/prompting-flow-designer.md` [121 lines] *Prompt Flow Panel: Open Panels → Prompt Flow Panel from chat, or File → Prompt Flow Panel from the Agentic Control Panel.*
F0295 `docs/self-management-carriage.md` [567 lines] *Self-modify and self-update carriage review: The combined builder predicts both editions' actual emitted version before building, pins one matching number…*
F0296 `docs/setup-lifecycle-research.md` [129 lines] *Windows setup: research and implementation decisions: Research date: 2026-10-11.*
F0297 `docs/setup-wizards.md` [257 lines] *Installation and removal windows: Status: shipped in the published v1.82.0 release (tag at 880a335e, 2026-10-12).*
F0298 `docs/stm32er_all_families_proposal.md` [171 lines] *Proposal — Widen STM32er to the ENTIRE ST 32-bit line (Blue Pill → STM32N6)*
F0299 `docs/terminater.md` [346 lines] *Terminater — visible, user-supervised terminals: Terminater is the 86th workflow agent and the first one whose result you can *watch*. Executer runs a command…*
F0300 `docs/torrenter.md` [114 lines] *Torrenter: Torrenter is a Tlamatini workflow agent and the wrapped chat tool chat_agent_torrenter.*
F0301 `docs/usage-dashboard.md` [222 lines] *About Usage: Status: shipped in the published v1.78.0 release (2026-10-10).*
F0302 `docs/user-creation-wizard.md` [99 lines] *Create a Tlamatini login account: Current version: v1.92.4 (b99ea506); last inspected at a771d27c on main on 2026-10-23 (tag ebc5c3b4; migration 0219 lands…*
F0303 `docs/visual-analysis-errors.md` [119 lines] *Image and video failures, dialogs and recovery: Image/video analysis uses the configured model identities.*
F0304 `docs/windows-flow-files.md` [213 lines] *Windows flow files: Implemented 2026-10-04; shipped in v1.76.0.*

docs/claude/ · assistant onboarding (auto-imported) — sheet 24

F0270 `docs/claude/acpx.md` [458 lines] *Tlamatini — ACPX (Agent Communication Protocol eXtension): This file is the authoritative reference for what ACPX means in Tlamatini, the mechanics it…*
F0271 `docs/claude/agents.md` [409 lines] *Tlamatini — Agents (Creation, Catalog, FlowCreator, FlowHypervisor): Torrenter adds account-owned downloads to chat and the Agentic Control Panel.*
F0272 `docs/claude/architecture.md` [941 lines] *Tlamatini — Architecture & Core Systems: The combined builder predicts both editions' actual emitted version before building, pins one matching number, runs…*
F0273 `docs/claude/exec-report.md` [350 lines] *Tlamatini — Exec Report (Per-Agent Execution Tables on the Final Answer)*
F0274 `docs/claude/frontend.md` [576 lines] *Tlamatini — Frontend Architecture: Torrenter adds account-owned downloads to chat and the Agentic Control Panel.*
F0275 `docs/claude/gotchas.md` [412 lines] *Tlamatini — Gotchas, Build/Lint, Roadmap, Work Style: The combined builder predicts both editions' actual emitted version before building, pins one matching…*
F0276 `docs/claude/INDEX.md` [156 lines] *docs/claude/ — Tlamatini AI-Assistant Docs Index: The combined builder predicts both editions' actual emitted version before building, pins one matching…*
F0277 `docs/claude/mcp-tools.md` [293 lines] *Tlamatini — Creating a New MCP or Tool: Torrenter adds account-owned downloads to chat and the Agentic Control Panel.*
F0278 `docs/claude/multi-turn.md` [394 lines] *Tlamatini — Multi-Turn Mode, Create Flow, Parametrizer Sections: Torrenter adds account-owned downloads to chat and the Agentic Control Panel.*
F0279 `docs/claude/recent-fixes.md` [5.7k lines] *Tlamatini — Recent Fixes / Gotchas (archived fix log): The combined builder predicts both editions' actual emitted version before building, pins one matching…*

docs/changes/ · dated change records — sheet 24

F0240 `docs/changes/2026-10-03-prompt-commentary-input.md` [71 lines] *2026-10-03 — Prompt Flow commentary/input separation: Source changes, uncommitted.*
F0241 `docs/changes/2026-10-03-prompt-commentary-input.patch` [1.4k lines] *Patch recorded together with the change note 2026-10-03-prompt-commentary-input.*
F0242 `docs/changes/2026-10-03-prompt-flow-documentation.md` [59 lines] *2026-10-03 — Prompt Flow documentation and process cleanup: Ok, now update all all all all all all, all the related hundreds of documents, all the mardown…*
F0243 `docs/changes/2026-10-03-prompt-flow-documentation.patch` [631 lines] *Patch recorded together with the change note 2026-10-03-prompt-flow-documentation.*
F0244 `docs/changes/2026-10-04-commentary-graphical-redesign.md` [55 lines] *2026-10-04 — Graphical rich-text User Commentary: Sorry, but still horrible your fucking User Commentary: -It cannot be resized, directly in the borders of…*
F0245 `docs/changes/2026-10-04-commentary-graphical-redesign.patch` [2.0k lines] *Patch recorded together with the change note 2026-10-04-commentary-graphical-redesign.*
F0246 `docs/changes/2026-10-04-prompt-flow-documentation.md` [43 lines] *Prompt Flow documentation and dossier refresh — 2026-10-04: The standing instruction is do not commit.*
F0247 `docs/changes/2026-10-04-release-validation.md` [171 lines] *Release validation and repairs — October 4, 2026: This is an uncommitted local release campaign against source and Windows frozen artifacts reporting version…*
F0248 `docs/changes/2026-10-04-run-output-resize.md` [36 lines] *Prompt Flow Run output resizing — 2026-10-04: The horizontal divider resizes Run output from 5% to 95% of the combined canvas/output pane height; fixed…*
F0249 `docs/changes/2026-10-04-run-output-resize.patch` [834 lines] *Patch recorded together with the change note 2026-10-04-run-output-resize.*
F0250 `docs/changes/2026-10-04-windows-flow-files.md` [51 lines] *Windows flow files — 2026-10-04 source implementation: The existing format is .flw (agent flows), not .fmt.*
F0251 `docs/changes/2026-10-07-latexer-verification.md` [60 lines] *LaTeXer delivery and real-chat verification — 2026-10-07: The installed Tlamatini chat produced a correct simple PDF through LaTeXer…*
F0252 `docs/changes/2026-10-07-pdf-mover-incident.md` [87 lines] *2026-10-07: installed PDF delivery and Desktop deletion incident: The installed application's log records LaTeXer delivering…*
F0253 `docs/changes/2026-10-07-windows-access-verification.md` [82 lines] *Windows access scripts and non-admin execution checks — 2026-10-07: The security/ changes prepare Windows exceptions for the operator to apply.*
F0254 `docs/changes/2026-10-10-composer-resize.md` [80 lines] *Composer resizing and viewport containment — 2026-10-10: Status: shipped in the published v1.78.0 release (commit 113eddc4, 2026-10-10).*
F0255 `docs/changes/2026-10-11-ui-sweep.md` [146 lines] *Whole-site appearance and resize sweep — 2026-10-11: Local source changes requested by Angela.*
F0256 `docs/changes/2026-10-13-docked-avatar-divider.md` [65 lines] *Docked avatar resizing from the composer divider: Date: 2026-10-13.*
F0257 `docs/changes/2026-10-16-v1.85.0.md` [99 lines] *v1.85.0 — Release record (2026-10-16): Tlamatini v1.85.0 was the latest released version on 2026-10-16 (superseded by v1.87.0).*
F0258 `docs/changes/2026-10-18-v1.87.0.md` [128 lines] *v1.87.0 — Release record: Tlamatini v1.87.0 was tagged on 2026-10-18.*
F0259 `docs/changes/2026-10-19-private-release-data.md` [97 lines] *2026-10-19 — The private release carries the data of the machine that built it*
F0260 `docs/changes/2026-10-19-terminater.md` [103 lines] *2026-10-19 — Terminater: visible, user-supervised terminals: Status: released in v1.88.0 (tagged 2026-10-20), carried in v1.91.1.*
F0261 `docs/changes/2026-10-20-terminater-audit.md` [181 lines] *Terminater audit — 2026-10-20: This is a source-checkout audit and a set of demonstrated corrections, not a certificate that Tlamatini has no bugs.*
F0262 `docs/changes/2026-10-20-v1.88.0.md` [94 lines] *2026-10-20 — v1.88.0: Terminater and a private release that carries your data*
F0263 `docs/changes/2026-10-22-documentation-refresh.md` [76 lines] *Documentation refresh - v1.90.0 (2026-10-22): Requested after Angela tagged, committed and pushed the implementation.*
F0264 `docs/changes/2026-10-22-release-version-pinning.md` [73 lines] *Release version pinning — 2026-10-22: Angela reported a public release ending with build_installer.py failed. after configuration restoration.*
F0265 `docs/changes/2026-10-22-v1.90.0.md` [98 lines] *v1.90.0 — Live editor tutoring and precise user intent: Documentation baseline: 2026-10-22, America/Mexico_City.*
F0266 `docs/changes/2026-10-23-v1.91.0.md` [148 lines] *v1.91.0 — Configure how Tlamatini types, teaches and speaks: v1.91.0 was the release of 2026-10-23; current guidance is v1.92.3.*
F0267 `docs/changes/2026-10-27-v1.91.1.md` [51 lines] *v1.91.1 — Prompt Flow fixes and short tool names: Release on this date: v1.91.1, dated 2026-10-27.*
F0268 `docs/changes/2026-11-01-v1.92.3.md` [26 lines] *Tlamatini 1.92.3: Historical release: 1.92.3.*

authoring guides — sheet 24

F0471 `Tlamatini/.agents/workflows/create_new_agent.md` [753 lines] *How to create a new Tlamatini agent*
F0472 `Tlamatini/.mcps/create_new_mcp.md` [888 lines] *How to add a new MCP-backed context provider or unified-agent tool to Tlamatini without confusing the transport, DB, UI, and chain layers.*
F0473 `Tlamatini/.skills/create_new_skill.md` [416 lines] *Creating a New Skill (SKILL.md package) in Tlamatini: Authoritative, step-by-step guide for authoring a new Skill — including ACPX skills.*

scripts/architecture_atlas/ and this atlas — sheet 24

F0340 `scripts/architecture_atlas/build_architecture_pdf.py` [238 lines] *Rebuild Tlamatini-Architecture.pdf in one step.*
F0341 `scripts/architecture_atlas/gen_architecture.py` [2.8k lines] *Generate Tlamatini-Architecture.tex: a multi-sheet architecture atlas of Tlamatini.*
F0342 `scripts/architecture_atlas/README.md` [54 lines] *Architecture atlas — how to rebuild Tlamatini-Architecture.pdf: Tlamatini-Architecture.pdf (repository root) is a multi-sheet A3 architecture atlas of…*
F0466 `Tlamatini-Architecture.pdf` [2.1MB] *This architecture atlas: every listed application file on A3 sheets; rebuilt by scripts/architecture_atlas/build_archi… and typeset by LaTeXer.*

.claude/ · Claude Code skills, hooks and settings — sheet 24

F0001 `.claude/hooks/announce_skills.py` [204 lines] *SessionStart hook – announce that the complete skill set (Claude Code skills + Tlamatini's SKILL.md packages) is loaded, so every session is…*
F0002 `.claude/hooks/test_announce_skills.py` [353 lines] *Automated tests for the SessionStart hook (announce_skills.py). [15 tests]*
F0086 `.claude/settings.json` [16 lines] *JSON document with keys: hooks.*
F0087 `.claude/skills/tlamatini-agent-creation/SKILL.md` [958 lines] *The authoritative, exhaustive end-to-end runbook for creating a BRAND-NEW Tlamatini workflow agent — every surface, in order, with 530+ numbered…*
F0088 `.claude/skills/tlamatini-agent-naming/SKILL.md` [100 lines] *The authoritative Tlamatini agent NAMING CONVENTION — invoke before adding, renaming, casing, displaying, or auditing any workflow agent (STM32er…*
F0089 `.claude/skills/tlamatini-daily-chat-test/.gitignore` [8 lines] *Per-run outputs (regenerated daily)*
F0151 `.claude/skills/tlamatini-daily-chat-test/SKILL.md` [278 lines] *Run the daily automated Tlamatini chat regression — drive a visible Chrome via Playwright, log into agent_page.html, ask up to 1000 curated safe…*
F0152 `.claude/skills/tlamatini-self-modify-inclusion/scripts/sweep_self_modify.py` [436 lines] *sweep_self_modify.py – deterministic auditor for Tlamatini's self-MODIFY snapshot.*
F0153 `.claude/skills/tlamatini-self-modify-inclusion/SKILL.md` [354 lines] *Sweep the whole codebase and keep the SELF-MODIFY source snapshot complete — so the TlamatiniSourceCode/ tree that a –self-modify build ships…*
F0154 `.claude/skills/tlamatini-self-update-inclusion/scripts/sweep_self_update.py` [580 lines] *sweep_self_update.py – deterministic auditor for Tlamatini's self-update pipeline.*
F0155 `.claude/skills/tlamatini-self-update-inclusion/SKILL.md` [441 lines] *Sweep the whole codebase and keep the SELF-UPDATE pipeline complete — so every new asset/feature added to Tlamatini is actually carried into a…*

.claude/memory/ · persistent assistant memory — sheet 24

F0003 `.claude/memory/feedback_agent_naming_conventions.md` [33 lines] *name: feedback_agent_naming_conventions description: CRITICAL — Tlamatini agent naming convention.*
F0004 `.claude/memory/feedback_developer_product_boundary.md` [18 lines] *Developer workflow is not product UX: Angela clarified this on 2026-09-27: she is the developer using this machine; the worldwide audience for her products is…*
F0005 `.claude/memory/feedback_dont_overbuild_exec_safety.md` [29 lines] *The whole multi-day external-exec safety hardening (gates/containment/_run_bounded/watcher daemon) was discarded by the user as not-working — don't…*
F0006 `.claude/memory/feedback_forbidden_headless_visible_tests.md` [20 lines] *All automated tests and diagnostics must remain visible on Angela's real desktop, with no headless exception.*
F0007 `.claude/memory/feedback_foreground_visible_windows.md` [19 lines] *Every Tlamatini execution must run in a visible forked foreground window; headless is forbidden.*
F0008 `.claude/memory/feedback_hard_real_scenario_tests.md` [25 lines] *User demands HARD, real-scenario automated tests — soft happy-path tests are unacceptable*
F0009 `.claude/memory/feedback_main_branch_only.md` [40 lines] *ABSOLUTE rule. Tlamatini is a solo project; NEVER create branches, NEVER suggest PRs, NEVER let a "Create PR" UI badge appear. Everything goes on…*
F0010 `.claude/memory/feedback_package_json_version_bump.md` [32 lines] *On release version bumps, also bump package.json \"version\" — user wants it tracked despite the git-tag-only contract*
F0011 `.claude/memory/feedback_run_tlamatini_agents_visible.md` [34 lines] *To show Tlamatini's desktop agents (Playwrighter headed, forked windows, Shoter/Mouser/Keyboarder) on the user's real screen, launch FOREGROUND with…*
F0012 `.claude/memory/feedback_state_constraints_upfront.md` [31 lines] *Surface hard platform/architectural constraints + real options UP FRONT as a decision for the user — never discover them by trial-and-error.*
F0013 `.claude/memory/feedback_track_changes_pivot_file.md` [27 lines] *Keep a pivot/changelog file recording the verbatim request + complete before/after of every change, so precise rollbacks never depend on git or on…*
F0014 `.claude/memory/feedback_update_agent_docs.md` [27 lines] *When adding or modifying workflow agents in Tlamatini, also update the agent documentation files*
F0015 `.claude/memory/feedback_user_owns_git.md` [43 lines] *Read-only git commands (status/log/diff/show/blame) are always allowed.*
F0016 `.claude/memory/MEMORY.md` [151 lines] *v1.92.4 — Matching builds, explicit release identity and readable Markdown*
F0017 `.claude/memory/project_acpx_oneshot_prompt.md` [46 lines] *2026-05-04 fix for ACPX losing TUI agent responses; claude/gemini/cursor/qwen/codex now re-spawn per turn with prompt as CLI arg*
F0018 `.claude/memory/project_acpx_skills_menu.md` [58 lines] *2026-05-17 — Added \"ACPX-Skills\" navbar dropdown (Browse / Configure / Diagnostics / Reload).*
F0019 `.claude/memory/project_acpx_toggle_skills.md` [35 lines] *2026-05-06 - acpx_enabled WebSocket flag filters the 12-tool ACPX/Skill surface per-request via agent.acpx.filter_acpx_tools(); 21 SKILL.md packages…*
F0020 `.claude/memory/project_acpxer_added.md` [31 lines] *2026-05-03 added ACPXer (#59) — a workflow agent that drives one ACPX session lifecycle from the canvas; it deliberately mirrors the runtime…*

F0021 `.claude/memory/project_agent_table_wiped_on_boot.md` [71 lines] *apps.ready() DELETES the whole Agent table every startup and re-derives names with .title() — the real source of 'Pdfer' and 6 dead canvas…*
F0022 `.claude/memory/project_arduiner_agent.md` [35 lines] *Arduiner (#70) Arduino CLI bridge agent — built 2026-06-02, full canvas + Multi-Turn, ships ArduinoTemplateProject; not committed*
F0023 `.claude/memory/project_ask_execs_feature.md` [35 lines] *The \"Ask Execs\" per-tool Proceed/Deny permission prompt for Multi-Turn executions — architecture + contracts.*
F0024 `.claude/memory/project_audioplayer_agent.md` [31 lines] *AudioPlayer (#73) audio-playback workflow agent — soundfile+sounddevice, canvas + chat_agent_audioplayer*
F0025 `.claude/memory/project_build_concurrency_guard.md` [32 lines] *build.py error 'FileNotFoundError build/manage/warn-manage.txt' = TWO concurrent builds clobbering shared build/ dist/ work dirs; fixed with a PID…*
F0026 `.claude/memory/project_build_tests_and_installer_enterkey.md` [25 lines] *build-script contract tests (test_build_scripts.py, 28) + Enter-key triggers install/uninstall in the Tkinter GUIs*
F0027 `.claude/memory/project_camcorder_agent.md` [27 lines] *2026-06-03 added Camcorder (#71) webcam photo/video agent (canvas + chat_agent_camcorder)*
F0028 `.claude/memory/project_carried_python_for_agents.md` [37 lines] *Frozen installer now ships its OWN Python 3.12.10 for pool agents; agents ALWAYS use it (no system Python needed)*
F0029 `.claude/memory/project_conjunction_parser_fix.md` [26 lines] *2026-04-24 fix — _split_assignment_segments and _closes_outer_quote now split on and KEY= / with KEY=, not just ,/;. Root-cause of AngysBackInCUDA…*
F0030 `.claude/memory/project_daily_chat_test_skill.md` [28 lines] *The tlamatini-daily-chat-test Claude Code skill — Playwright/real-Chrome harness that asks 1000 curated questions to the Tlamatini chat daily and…*
F0031 `.claude/memory/project_desktop_ui_lifecycle.md` [29 lines] *2026-05-07 fix — Keyboarder/Mouser/Shoter purposes + prompt.pmt rule 11 codify open→focus→interact→close→handle-save-…*
F0032 `.claude/memory/project_director_virtuoso_demo_prompts.md` [28 lines] *2026-05-21 added two ADVANCED multi-agent demo prompts (slots 55/56) via migration 0096; catalog now runs 1-56, next free slot is 57*
F0033 `.claude/memory/project_doc_refresh_2026_05_06.md` [40 lines] *Comprehensive doc-refresh pass covering the last week of changes - ACPX toggle, Skills system, WhatsTlamatini, ACPXer, summarizer one-shot mode…*
F0034 `.claude/memory/project_doc_refresh_2026_05_08.md` [39 lines] *Cross-cutting doc audit and surgical fixes to CLAUDE.md, docs/claude/exec-report.md, docs/claude/frontend.md, and README.md to align with the last 2…*
F0035 `.claude/memory/project_doc_refresh_2026_05_09.md` [58 lines] *CLAUDE.md + docs/claude/* updated for Flow Compiler / Agent Contracts (commit 0bea21d), agents_descriptions.md authoritative tooltip source…*
F0036 `.claude/memory/project_eight_fixes_2026_05_07.md` [53 lines] *2026-05-07 batch fixing the "open notepad → type → wait → close" failure mode (15 iterations got force-stopped before Keyboarder ran)*
F0037 `.claude/memory/project_emailer_recmailer_oneshot.md` [36 lines] *Emailer/RecMailer now work standalone in Multi-Turn chat (one-shot send / one-shot check)*
F0038 `.claude/memory/project_embedding_memory_guard.md` [35 lines] *2026-05-12 — Embedding-memory pre-flight guard added (NVIDIA-GPU-only, fail-open) for the \"Set directory as context\" flow*
F0039 `.claude/memory/project_esp32er_agent.md` [37 lines] *ESP32er agent (#69) — PlatformIO pio CLI bridge, the direct-CLI sibling of STM32er*
F0040 `.claude/memory/project_execute_command_bounded_fix.md` [27 lines] *2026-06-06 execute_command now bounded (timeout + stdin=DEVNULL + kill-tree); fixes the copy-con hang*
F0041 `.claude/memory/project_execute_file_foreground_fix.md` [44 lines] *REAL root cause of 'execute_file window never opened + false OK'; 2026-05-29 made the foreground path reachable, 2026-05-30 fixed the LAUNCH…*
F0042 `.claude/memory/project_external_exec_safety_layer.md` [33 lines] *The Python-launch syntax gate + Job-Object containment safety layer, its coverage, and the outstanding Sqler/Mongoxer gap*
F0043 `.claude/memory/project_flow_compiler_dialog_wins.md` [29 lines] *Reverses the canvas-as-source-of-truth contract for connection fields; user-edited source_agents/target_agents/etc. survive Start.*
F0044 `.claude/memory/project_ghost_distinfo_dependency_audit.md` [56 lines] *site-packages carries ~30 GHOST dist-info dirs; pip's conflicts were mostly phantom. audit_dependencies.py + the numpy floor fix (2026-07-26)*
F0045 `.claude/memory/project_kalier_agent.md` [37 lines] *Kalier agent — Kali Linux / MCP-Kali-Server bridge added on canvas + Multi-Turn*
F0046 `.claude/memory/project_keyboarder_wrapped_2026_05_07.md` [32 lines] *2026-05-07 — Keyboarder + Shoter usable from Multi-Turn; chat_agent_keyboarder added to registry, _EXEC_REPORT_TOOLS, CSS, prompt; migration 0078*
F0047 `.claude/memory/project_latexer_agent.md` [76 lines] *LaTeXer (#87) — native typesetting, 30 optional styles, engine-free catalogue, MiKTeX-backed compilation and portable pool helpers.*
F0048 `.claude/memory/project_living_canvas_dropped.md` [29 lines] *2026-04-29 — Living Canvas v0 commits (2fc441d, 0705a9c, d496cda) were force-reverted; feature was non-functional in production despite passing…*
F0049 `.claude/memory/project_mouser_wrapped_2026_05_07.md` [37 lines] *2026-05-07 — Mouser usable from Multi-Turn; chat_agent_mouser added to registry, _EXEC_REPORT_TOOLS, CSS, prompt; migration 0079.*
F0050 `.claude/memory/project_native_picker_tcltk_fix.md` [57 lines] *Server file picker rewritten tkinter-free (Win32 ctypes); tkinter REMOVED from server build.*
F0051 `.claude/memory/project_native_toast.md` [31 lines] *Notifier toast/desktop-popup experiment was REMOVED entirely (never worked); only the Windows "Installed apps" registration was kept.*
F0052 `.claude/memory/project_netspeed_calculator_agent.md` [89 lines] *NetSpeed-Calculator (agent #88) — COMPLETE and wired across ~30 surfaces; the 4 endpoint bugs, the silent-zero fix, and the spaced-impostor Agent…*
F0053 `.claude/memory/project_numpy_pyinstaller.md` [24 lines] *Root cause and fix for "cannot load module more than once per process" at Tlamatini startup*
F0054 `.claude/memory/project_ollama_source_build_breaks_embeddings.md` [31 lines] *The \"llama-server binary not found\" / Nomic-Embed error is the broken C:\\Development\\ollama source build, NOT prompt size*
F0055 `.claude/memory/project_pdfer_agent.md` [78 lines] *PDFer document composer — 24 visual styles, 20 semantic themes, measured layouts, catalog mode and flow outputs*
F0056 `.claude/memory/project_planner_followup.md` [31 lines] *Fixed planner losing tool context on short follow-up messages, plus 5 other fixes from MXNet installation log analysis*
F0057 `.claude/memory/project_playwrighter_agent.md` [31 lines] *Playwrighter (#65) scripted-browser-automation agent — canvas + Multi-Turn, added 2026-05-20*
F0058 `.claude/memory/project_playwrighter_hold_open.md` [56 lines] *Playwrighter hold_open_seconds/hold_open_ms linger knob — fixes \"browser closes before I can see it*
F0059 `.claude/memory/project_prompt_catalog_mode_badges.md` [42 lines] *Prompt catalog cards now show mode badges (One-Shot / Multi-turn / ACPX) and auto-set the Multi-Turn + ACPX toolbar checkboxes when a prompt is…*
F0060 `.claude/memory/project_prompt_flow_commentary_input.md` [19 lines] *Preserve the annotation/input split, mixed-style graphical editing, file fidelity, 5–95% Run output resizing and source/release boundaries.*
F0061 `.claude/memory/project_pythonxer_forkbomb_fix.md` [46 lines] *Pythonxer crash (thousands of flashing console windows, machine restart) — fix = blocking syntax gate + blocking Ruff (non-zero on lint fail) +…*
F0062 `.claude/memory/project_pythonxer_strict_ruff_gate.md` [36 lines] *STRICT Pythonxer correctness gate (compile() floor + blocking Ruff) + LLM fix→re-ruff→retry loop — the sanctioned minimal subset re-added 2026-05-29…*
F0063 `.claude/memory/project_quote_apostrophe_fix_2026_05_07.md` [48 lines] *2026-05-07 fix for the "Hi!, i"m tlamatini" bug — the LLM passed "Hi!, I"m Tlamatini" (doubled outer + SQL-doubled inner apostrophe) and it…*
F0064 `.claude/memory/project_reaper_console_window_fix.md` [36 lines] *Orphan reaper was killing Tlamatini's own console-window conhost (frozen onedir builds), closing the window and hanging the server; fixed + tests*
F0065 `.claude/memory/project_recorder_agent.md` [31 lines] *Recorder (#72) microphone/audio-capture agent added on canvas + Multi-Turn*
F0066 `.claude/memory/project_reviewer_analyzer_agents.md` [33 lines] *2026-05-19 — added Reviewer + Analyzer workflow agents (#63/#64) and code-review + security-audit SKILL.md packages, on both surfaces*
F0067 `.claude/memory/project_reviewer_analyzer_demo_prompts.md` [25 lines] *2026-05-20 added two fancy seeded demo prompts for Reviewer/Analyzer (idPrompt 26/27) via migration 0090; documents the prompts-catalog contiguity…*
F0068 `.claude/memory/project_reviewer_committed_secrets_falsepos.md` [31 lines] *2026-05-20 fix — Reviewer agent + code-review skill falsely reported working-tree API keys as \"committed\"; taught both commit-state precision +…*
F0069 `.claude/memory/project_secret_leak_recovery.md` [33 lines] *How Tlamatini commits can leak API keys, and the recovery recipe when it happens*
F0070 `.claude/memory/project_stm32er_agent.md` [35 lines] *STM32er (#68) agent bridging the STM32TemplateProjectMCP stdio server on canvas + Multi-Turn*
F0071 `.claude/memory/project_stm32er_bootstrap_preflight.md` [31 lines] *STM32er zero-config auto-bootstrap (download/install MCP) + critical-mission safety preflight (validate compiler/CubeIDE/driver/ST-LINK, fail-safe…*
F0072 `.claude/memory/project_stm32er_demo_prompts.md` [29 lines] *STM32er Catalog-of-Prompts demos — migration 0103 seeds 3 tiered STM32F firmware prompts at slots 63/64/65*
F0073 `.claude/memory/project_stm32er_docs_libs_parametrizer.md` [35 lines] *STM32er v1.9.0 alignment — docs/skills/prompt.pmt swept for bootstrap+preflight, parametrizer server_script field, requirements lib sweep (PyPDF2…*
F0074 `.claude/memory/project_stm32er_hil_serial_proof_fix.md` [27 lines] *STM32er HIL demo (#65) fix — F4-Discovery ST-LINK VCP is NOT routed to the MCU USART, so live-SWD-memory is the real HIL proof, not serial*
F0075 `.claude/memory/project_stm32er_tests.md` [27 lines] *STM32er agent test suite — agent/test_stm32er_agent.py, 85 tests covering compiler-find/compile/link/hex via a fake MCP stdio server*
F0076 `.claude/memory/project_summarizer_oneshot_mode.md` [49 lines] *2026-05-06 fix — chat_agent_summarize_text now actually accepts input_text/target_words; before this, the registry advertised it as a one-shot text…*
F0077 `.claude/memory/project_taskbar_flash_attention.md` [30 lines] *Taskbar-flash + uppercase log banner attention notice for Ask-Execs prompts and Notifier notifications (the toast successor)*
F0078 `.claude/memory/project_teletlamatini_acpx.md` [34 lines] *TeleTlamatini AND WhatsTlamatini carry acpx_enabled end-to-end (config default true); 2026-06-04 both also hardened for "Ask Execs" — ungated by…*
F0079 `.claude/memory/project_temp_templates_policy.md` [39 lines] *Temp + Templates dirs must live ONLY under the Tlamatini app root; never outside Tlamatini*
F0080 `.claude/memory/project_unity_mcp_declined.md` [30 lines] *2026-05-17 — Uniter agent (Unity MCP analogue of Unrealer) was analyzed but explicitly NOT built; functionality too limited to justify*
F0081 `.claude/memory/project_unrealer_expanded_surface.md` [101 lines] *Unrealer agent expanded from 28→53 Unreal-MCP commands (System/Level/Asset/Material); cross-repo fork relationship + forwarder hardening*
F0082 `.claude/memory/project_versioning_2026_05_15.md` [71 lines] *SemVer 2.0.0 versioning system with git-tag-derived versions, injected into build artefacts at build time.*
F0083 `.claude/memory/project_videoplayer_agent.md` [33 lines] *VideoPlayer (#74) video+audio playback agent — ffpyplayer+OpenCV, canvas + chat_agent_videoplayer*
F0084 `.claude/memory/project_windower_agent.md` [37 lines] *Windower (#65) Win32 window-manager agent — dual-surface (canvas + chat_agent_windower), ports Windows-MCP window subset*
F0085 `.claude/memory/user_profile.md` [18 lines] *Primary developer of Tlamatini; values robustness, comprehensive execution, and parallel work.*

.gemini/skills/ · Gemini mirror — sheet 24

F0161 `.gemini/skills/tlamatini-agent-creation/SKILL.md` [958 lines] *The authoritative, exhaustive end-to-end runbook for creating a BRAND-NEW Tlamatini workflow agent — every surface, in order, with 530+ numbered…*
F0162 `.gemini/skills/tlamatini-agent-naming/SKILL.md` [100 lines] *The authoritative Tlamatini agent NAMING CONVENTION — invoke before adding, renaming, casing, displaying, or auditing any workflow agent (STM32er…*
F0163 `.gemini/skills/tlamatini-daily-chat-test/.gitignore` [8 lines] *Per-run outputs (regenerated daily)*
F0196 `.gemini/skills/tlamatini-daily-chat-test/SKILL.md` [278 lines] *Run the daily automated Tlamatini chat regression — drive a visible Chrome via Playwright, log into agent_page.html, ask up to 1000 curated safe…*
F0197 `.gemini/skills/tlamatini-self-modify-inclusion/scripts/sweep_self_modify.py` [436 lines] *sweep_self_modify.py – deterministic auditor for Tlamatini's self-MODIFY snapshot.*
F0198 `.gemini/skills/tlamatini-self-modify-inclusion/SKILL.md` [354 lines] *Sweep the whole codebase and keep the SELF-MODIFY source snapshot complete — so the TlamatiniSourceCode/ tree that a --self-modify build ships…*
F0199 `.gemini/skills/tlamatini-self-update-inclusion/scripts/sweep_self_update.py` [580 lines] *sweep_self_update.py – deterministic auditor for Tlamatini's self-update pipeline.*
F0200 `.gemini/skills/tlamatini-self-update-inclusion/SKILL.md` [441 lines] *Sweep the whole codebase and keep the SELF-UPDATE pipeline complete — so every new asset/feature added to Tlamatini is actually carried into a…*

.codex/skills/ · Codex dossier and release skills — sheet 24

F0156 `.codex/skills/full-project-pdf-dossier/SKILL.md` [150 lines] *Use when asked to create or update a complete project PDF dossier, app summary, architecture report, or repository documentation PDF that must…*
F0157 `.codex/skills/overlap-safe-pptx-dossier/SKILL.md` [161 lines] *Use when asked to create or update a complete technical PPTX deck, especially a Tlamatini-style presentation, where slides must describe the whole…*
F0158 `.codex/skills/tlamatini-release-validation/agents/openai.yaml` [4 lines] *YAML document: keys interface.*
F0159 `.codex/skills/tlamatini-release-validation/references/acceptance-matrix.md` [111 lines] *Acceptance matrix: Apply this to the current build, using actual UI labels and endpoints found in source.*
F0160 `.codex/skills/tlamatini-release-validation/SKILL.md` [120 lines] *Validate a real Tlamatini Windows release through clean installation, administration, chat, Prompt Flow Panel, Agentic Control Panel, and lifecycle…*