AN

NLTlabs

NEW LOGIC TECH

NLT Quality Pipeline

Builder Ops · dev-time gates & docs

Quality MCP · docs MCP · gates · judges · skills.

26 curated MCP tools across 7 families in this volume.
Dev-time scoring and checklist — distinct from runtime NLT Memory (Vol 8).
Consumers bootstrap via tapps_init; document repos add validate_changed judges.
Repo paths and CLI names stay internal; outward suite uses NLT component names.

AUDIENCE Agent authors · platform maintainers · consumer repo operators

Contents

How to read this. Maintainer ops for the NLT Quality Pipeline MCP servers (tapps-mcp, docs-mcp checkouts at /home/wtthornton/code/tapps-mcp when present). For memory tiers, bridge flow, and hive propagation see **Vol 8 NLT Memory**. For report-studio CLI see **Vol 9**.

Part I — Pipeline role and boundaries

PART I

NLT Quality Pipeline at a glance

NLT Quality Pipeline is the dev-time quality and documentation layer for AI coding assistants. The tapps-mcp server exposes deterministic tools — scoring, security scans, quality gates, checklist enforcement, and Linear cache wrappers. docs-mcp generates README, changelog, and validated Linear issue bodies. Neither server embeds portfolio logic; consumer repos (ReportLab, NLTlabsPE, AgentForge) pin versions and configure judges for their output shape.

Three-way distinction

Layer	Volume	Runtime?
NLT Memory (brain HTTP API)	Vol 8	Yes — agents recall during work
Report Studio (PDF engine)	Vol 9	Build time — briefs.nltlabs.ai
NLT Quality Pipeline (this volume)	Vol 10	Dev time — IDE MCP session only

> "Agents must not conflate NLT Quality Pipeline session hooks with the federated NLT Memory service — bridge HTTP only, never in-process brain embed."
>
> **— Consumer boundary**

Part II — Monorepo package map

PART II

Packages and consumer wiring

Path	Role
packages/tapps-core	Shared config, metrics, brain bridge, judges
packages/tapps-mcp	MCP server — scoring, gates, checklist, Linear wrappers
packages/docs-mcp	README/changelog/Linear doc generators (companion)
packages/tapps-platform	Combined serve entry for 47+ tool hosts
.tapps-mcp.yaml	Consumer preset: engagement, judges, memory profile
.cursor/skills/tapps-*	Finish-task, handoff, validate bundles
.github/workflows/tapps-quality.yml	PR gate on changed Python

Bootstrap path. tapps_init writes AGENTS.md, TECH_STACK.md, platform hooks, skills, and optional .mcp.json bridge config. with_report_studio=True adds report-studio consumer scaffold. Brain entries strip direct tapps-brain — HTTP bridge only.

Part III — Pipeline stages

PART III

NLT quality pipeline

Every session follows five stages. Skipping research (lookup_docs) is the most common source of hallucinated APIs; skipping verify (checklist) is the most common source of undeclared work.

Stage	Primary tools	Done when
discover	tapps_session_start, tapps_server_info	Checker env + brain health known
research	tapps_lookup_docs, tapps_memory (Vol 8)	APIs verified before coding
develop	tapps_score_file, tapps_quick_check	Iterative gate on edits
validate	tapps_validate_changed, tapps_security_scan	Batch + judges pass
verify	tapps_checklist, tapps_memory save	No required steps skipped

Preset tool bundles

Role presets (reviewer, developer, planner) and mode presets (quality, admin) trim MCP tool surface for Cursor's 40-tool limit. Full canonical list lives in server.py::ALL_TOOL_NAMES (38 tools at 3.12.x).

Part IV — Tool families

PART IV

MCP tool surface

Overview of 26 tools in 7 families. Each family follows with full parameter guidance.

Family	Tools	Theme
Session bootstrap	session_start, server_info, doctor, usage	First calls in every IDE session — server version, checker env, brain_bridge_health.
Scoring and gates	quick_check, score_file, quality_gate, validate_changed	Deterministic Python quality — ruff, mypy, bandit, radon, vulture, pylint, pip-audit.
Security and dependencies	security_scan, dependency_scan, dead_code, dependency_graph	Dedicated scans beyond quick_check basic depth.
Docs and impact	lookup_docs, impact_analysis, validate_config	Prevent hallucinated APIs and blind refactors.
Pipeline orchestration	checklist, init, upgrade, set_engagement_level	Checklist, init, upgrade, engagement — consumer repo scaffolding.
Linear cache-first reads	linear_snapshot_get, linear_snapshot_put, linear_list_issues, release_update	Write paths go through linear-issue skill; list reads use snapshot dance.
Audit campaigns	audit_campaign, audit_close_coverage, finding_to_story	Repo-wide quality campaigns and finding closure.

Session bootstrap

First calls in every IDE session — server version, checker env, brain_bridge_health.

Tool	Purpose	When
tapps_session_start	Project context + pipeline hint	FIRST call each chat
tapps_server_info	Version and installed checkers	Diagnostics without full bootstrap
tapps_doctor	MCP + brain + tool drift	After upgrade or auth failures
tapps_usage	Per-session tool gaps	When checklist reports missed steps

Scoring and gates

Deterministic Python quality — ruff, mypy, bandit, radon, vulture, pylint, pip-audit.

Tool	Purpose	When
tapps_quick_check	Score + gate + basic security	After each Python edit
tapps_score_file	Seven-category score	Deep review or CI report
tapps_quality_gate	Pass/fail vs preset threshold	Before declaring file done
tapps_validate_changed	Batch gate + optional judges	End of multi-file work

Security and dependencies

Dedicated scans beyond quick_check basic depth.

Tool	Purpose	When
tapps_security_scan	Bandit + secrets	Security task_type or sensitive edits
tapps_dependency_scan	pip-audit CVEs	Before release
tapps_dead_code	vulture unused symbols	Refactors and epic boundaries
tapps_dependency_graph	Import coupling	Large module changes

Docs and impact

Prevent hallucinated APIs and blind refactors.

Tool	Purpose	When
tapps_lookup_docs	Context7-backed library docs	Before any external API use
tapps_impact_analysis	Direct + transitive dependents	Before layout/engine API edits
tapps_validate_config	Docker, CI, YAML manifests	Infra or brand/template YAML

Pipeline orchestration

Checklist, init, upgrade, engagement — consumer repo scaffolding.

Tool	Purpose	When
tapps_checklist	Required tools by task_type	FINAL verification step
tapps_init	Bootstrap AGENTS.md, hooks, skills	New consumer repo
tapps_upgrade	Refresh scaffolding from latest	After tapps-mcp release
tapps_set_engagement_level	high / medium / low enforcement	Operator tuning

Linear cache-first reads

Write paths go through linear-issue skill; list reads use snapshot dance.

Tool	Purpose	When
tapps_linear_snapshot_get	Cache-first issue slice	Before list_issues
tapps_linear_snapshot_put	Populate cache after fetch	Immediately after list
tapps_linear_list_issues	Gate-checked list wrapper	When cache miss
tapps_release_update	Release note body from CHANGELOG	linear-release-update skill

Audit campaigns

Repo-wide quality campaigns and finding closure.

Tool	Purpose	When
tapps_audit_campaign	Batch audit orchestration	Quality sweeps
tapps_audit_close_coverage	Link fix SHA to finding	After audit fix lands
tapps_finding_to_story	Finding → Linear story	Audit remediation loop

Part V — Quality presets and engagement

PART V

Scoring presets and enforcement

Preset	Behavior
standard	Default gate — test coverage weighted on all Python
report_authoring	Lower coverage weight on reports/** narrative modules
strict	Higher thresholds for library/engine code

Engagement levels

tapps_set_engagement_level switches high / medium / low — changes required checklist tools and which hooks are wired.

Level	document task	feature task	Hooks
high	validate_changed + checklist	score_file + quality_gate + security_scan	10 lifecycle hooks; linear cache gate block mode optional
medium	validate_changed	score_file + quality_gate	8 hooks; linear cache gate warn; pre-commit optional
low	quality_gate (feature)	quick_check	SessionStart only; judges advisory

Checklist task types

task_type	Required	Recommended
feature	score_file, quality_gate	security_scan, memory
bugfix	score_file	quality_gate, security_scan
refactor	score_file, quality_gate	dead_code, dependency_graph
security	security_scan, quality_gate	score_file, dependency_scan
document	validate_changed	validate_config, lookup_docs, checklist
review	score_file, security_scan, quality_gate	checklist, dead_code
epic	checklist	score_file, quality_gate
release	release_update	dependency_scan

document task_type requires tapps_validate_changed — use explicit file_paths and configure blocking judges for PDF/HTML consumers. See Part VI for ReportLab reference config.

Part VI — Document output judges

PART VI

Gating document regressions

Python quality gates do not catch thin PDF pages, missing link annotations, or dropped --audit in site prebuild. Consumers add validate_changed.judges — pytest, grep, shell, or exists types.

validate_changed.judges — ReportLab reference

Document-shipping consumers gate **output quality**, not only Python scores. ReportLab configures shell + grep judges in .tapps-mcp.yaml:

Judge	Type	Blocks when
scripts/run-pdf-judge-tests.sh	shell	uv pytest PDF regression fails
build-pdfs.mjs contains --audit	grep	Prebuild drops post-build audit

when_changed globs avoid running pytest on README-only diffs. quality_preset: report_authoring prevents bogus coverage failures on reports/** narrative modules. memory.profile: document-builder recalls audit profiles and env roots.

Part VII — Hooks and agent skills

PART VII

Platform automation

Platform hooks and agent skills

tapps_init writes Cursor / Claude Code hooks at medium or high engagement. Skills bundle repeatable pipelines agents invoke by name.

Artifact	Trigger	Effect
tapps-after-edit.sh	PostToolUse Write/Edit	Remind tapps_quick_check
tapps-stop.sh	Session end	Telemetry if validate_changed skipped
/tapps-finish-task	Agent skill	validate_changed + checklist + memory
/tapps-handoff-session	Agent skill	session-handoff.md + session_end
linear-read	Agent skill	snapshot_get before list_issues
linear-issue	Agent skill	docs_validate before save_issue

Part VIII — CI and doctor

PART VIII

Shipping discipline

CI, doctor, and checker environment

Scoring runs seven checkers when installed in the MCP host environment: ruff, mypy, bandit, radon, vulture, pylint, pip-audit. The MCP server process may differ from the consumer project venv — use shell judges with uv run pytest when host pytest is missing.

Surface	Command	When
.github/workflows/tapps-quality.yml	validate-changed --quick on PR	Python changes
.githooks/pre-commit	validate-changed on staged .py	install_git_hooks: true
tapps-mcp doctor --quick	Brain + MCP without version drift	After upgrade
tapps-mcp doctor	Full checker version audit	Cold start / CI image build

Part IX — Operations

PART IX

Runbook and troubleshooting

Troubleshooting matrix

Symptom	Likely cause	Fix
judges_passed=false, pytest not found	MCP host lacks consumer venv	shell judge → uv run pytest script
brain_bridge_health.ok=false	Auth or brain down	TAPPS_BRAIN_AUTH_TOKEN; tapps doctor
Gate fail on reports/story.py only	Coverage weight on narrative	quality_preset: report_authoring
list_issues cache gate warn	Snapshot miss	linear-read skill or snapshot_put
PDF audit fails thin pages	Story sparse or wrong profile	build --audit with template profile

> “Code gate ≠ document gate. Python can score 100 while PDFs lose link annotations or drop --audit in prebuild — configure judges explicitly.”
>
> **— ReportLab quality pipeline evaluation (2026-06)**

Operator runbook

Daily agent session

1. tapps_session_start() → 2. tapps_lookup_docs before libraries → 3. tapps_quick_check per edit → 4. tapps_validate_changed(file_paths=...) → 5. tapps_checklist(task_type=...).

After tapps-mcp release

tapps-mcp upgrade --force --host auto → restart MCP host → tapps_doctor --quick → tapps_upgrade in consumer repos → verify judges still pass.

Document / PDF ship (ReportLab)

cd apps/docs && npm run build:pdfs (all volumes --audit) → node scripts/deploy-briefs.mjs → spot-check briefs.nltlabs.ai downloads.

Cross-volume links

Volume	Relationship
Vol 8 NLT Memory	Runtime recall — tapps_memory bridge (not in-process embed)
Vol 9 Report Studio	PDF engine — consumers pin nlt-report-studio tag
Vol 1–4	Stakeholder/reference PDFs built with report-studio --audit
tapps-mcp AGENTS.md	Living tool reference — 38+ MCP tools + memory actions

Colophon

Generated by reports.tapps_platform_ops in ReportLab. Tool families mirror tapps-mcp 3.12.x surface; verify against AGENTS.md in consumer repos for live tool count.

NLT Labs · New Logic Tech
NLT Quality Pipeline — Builder Ops
Issued 2026-06-11 · 26 tools · 7 families