

RockyGuard Library - Phase 4.1 Customer Onboarding Dry-Run Report

Version 1.2.0

Copyright (c) 2025-2026 [Legal Entity TBD]. All rights reserved.

Document: RockyGuard_Phase4_1_Report.txt
Library: RockyGuard C++17 License Library
Release: v1.2.0
Report date: 2026-04-24
Status: SUBMITTED for reviewer approval

Distribution: Internal and contracted reviewers only. This document reproduces forensic detail on documentation defects fixed during the dry-run and should not be distributed to customers.

Companion document: RockyGuard_Phase4_1_Plan.txt (revision 3, 2026-04-24, commit cd70644). This report is the execution record for that plan.

1. Executive Summary

2. Scope and Methodology

- 2.1. Plan under execution
- 2.2. Three-pass sub-agent orchestration
- 2.3. Isolation discipline
- 2.4. What this report covers / does not cover

3. Run 1: First-pass dry-run

- 3.1. Setup
- 3.2. Stage-by-stage record
- 3.3. Findings (13 gaps across all severities)
- 3.4. In-session resolutions
- 3.5. Verdict

4. Run 2: Delta re-run

- 4.1. Rationale
- 4.2. Setup
- 4.3. Resolutions confirmed
- 4.4. New findings
- 4.5. Verdict

5. Run 3: Final full re-run

- 5.1. Rationale

5.2. Setup

5.3. Stages 0 through 5: all PASS

5.4. Verdict

6. Consolidated Gap List and Resolutions

6.1. Resolved in v1.2.0 (in-zip at tag)

6.2. Deferred to v1.2.1

6.3. Deferred to v1.3

7. Late-cycle Defects Registered in the QA Report

7.1. GA-002 (BLOCKER): Quick Start linking procedure incomplete

7.2. GA-013 (HIGH): Release-CRT library cannot serve Debug consumers

8. Meta-findings and Process Improvements

9. Risk Assessment and Sign-off

Appendix A: Artifact SHA-256s (customer-use zip, pinned per run)

Appendix B: Source-commit cross-reference

Appendix C: Full gap-list table (first pass)

1. EXECUTIVE SUMMARY

Phase 4.1 of the v1.2.0 QA plan is a customer onboarding dry-run: a new engineer unfamiliar with the library receives the customer-use zip and follows Customer_Documentation.txt cover-to-cover, with no access to the development tree, exercising every documented Quick Start and integration path. The goal is to surface documentation and packaging defects that only a fresh reader can see.

The plan (RockyGuard_Phase4_1_Plan.txt, revision 3) was executed in three successive sub-agent passes, each by a different fresh general-purpose sub-agent via the Claude Agent SDK, each in an isolated OS temporary directory with no git history or network access to the dev tree.

Top-line result:

PASS --- READY TO SIGN OFF FOR v1.2.0

Dimension	Result
Sub-agent runs	3 (first pass, delta re-run, final full re-run)
Stages executed end-to-end	6 (Stage 0 pre-flight + Stages 1-5 per plan)
Total gaps surfaced	15 (13 first-pass GA-, 2 delta-pass DR2-)
BLOCKER / HIGH closed in v1.2.0	2 (GA-002 BLOCKER, GA-013 HIGH)
MEDIUM closed in v1.2.0	4 (GA-003, GA-004, GA-005, GA-009)
LOW closed in v1.2.0	2 (GA-007, DR2-001)
Deferred to v1.2.1	6 (GA-001 HIGH / DR2-002 duplicate; GA-006, GA-010, GA-011, GA-012 LOW; plus GA-013 medium-term Debug-CRT library ship)
Deferred to v1.3	1 (GA-008 LOW legal-entity)
New late-cycle defects registered in QA Report §4.2	2 (GA-002 as §4.2.7, GA-013 as §4.2.8)
Final-run new findings	0
Stage 5 Premium / floating path tested	YES (final run; checkout / heartbeat / release / seat cap all green)

No BLOCKER issue remains open. One HIGH issue (GA-001, the vendor-license onboarding section) is deferred to v1.2.1; this is an acceptable pass condition under the QA Plan because GA-001 concerns support-content adequacy, not a code or shipped-package defect, and any v1.2.0 customer reaches their vendor license through the sales channel rather than by reading the docs cold. Phase 4.1 signs off on the customer-use zip as shipped (SHA-256 6f1634be700282c30667f4e74626276ca554045b665fa73d517d7855b8684a49, pinned in archives/MANIFEST.txt). Only Phase 6 (legal / branding review of the license agreement and copyright line) remains pre-tag for v1.2.0.

2. SCOPE AND METHODOLOGY

2.1 Plan under execution

This report documents execution of RockyGuard_Phase4_1_Plan.txt revision 3 (commit cd70644, dated 2026-04-24). The plan defines six execution stages:

Stage	Name	Purpose
0	Pristine environment setup	Verify isolation: no .git, no dev-tree visibility, no network to internal hosts.
1	Read the package cold	End-to-end read of README.txt + Customer_Documentation.txt + Customer_API_Reference.txt.
2	Quick Start end-to-end	2.1 license_keygen; 2.2 vendor license handoff; 2.3 license_create; 2.4 license_verify; 2.5 rg_fingerprint.
3	Minimal library integration	3.1/3.2 write my_app.cpp, build per docs; 3.3 bundle and deploy.
4	Adversarial probes	4.1 wrong key path; 4.2 strip .sig; 4.3 tamper DLL; 4.4 past-date license; 4.5 wrong-machine license.
5	Premium-tier features	floating_server_example + floating_client_example end-to-end (Premium tier only).

The plan specifies pass criteria, gap-severity definitions (BLOCKER / HIGH / MEDIUM / LOW), and a deliverable format (walk-through log + gap list + verdict). See Plan §5 and §6.

2.2 Three-pass sub-agent orchestration

A single dry-run is insufficient once documentation fixes are applied: the fixing engineer becomes contaminated with the knowledge of what was changed, and cannot re-test the customer's view. Phase 4.1 was therefore executed as three independent passes, each with a fresh sub-agent that had no memory of prior passes:

Pass	Agent	Scope	Zip under test
1	general-purpose sub-agent #1	Full Stages 0-4 (Stage 5 time-boxed out)	efe8284526fa65f103fc e5d213d4fdbbfed54ea5 f028a5a0563119e7fc29dfc2 (pre-fix zip)
2	general-purpose sub-agent #2	Delta on Stages 1-3 (where docs had changed)	2b0bb957f3426a967fdc fae84a24951836bbf28a 18a9136ff906ae9d9de574fc (post-first-pass-fix zip)
3	general-purpose sub-agent #3	Full Stages 0-5 clean run	6f1634be700282c30667 f4e74626276ca554045b 665fa73d517d7855b8684a49 (post-delta-pass-fix zip)

Each agent was briefed with the plan text plus pre-flight instructions. Each worked in a fresh plain-OS temp directory (C:\tmp\phase4_1_*) with no VCS metadata at the directory or any ancestor, and no shell history from prior passes. Every pre-flight verification (no .git, git status fails, no dev-tree files) must succeed before the stage clock starts.

2.3 Isolation discipline

The plan forbids a git worktree as the operator environment (R2-2 in the plan's revision history): a worktree carries a .git pointer file that silently grants access to the full development history, defeating the cold-read premise. All three passes therefore ran in a plain mkdir-created temp directory. Verification was built into Stage 0: the sub-agent ran git rev-parse --show-toplevel inside its cwd and required failure, listed .git* entries in every ancestor up to the drive root and required none, and attempted to open D:\license_checking\src\license.cpp and required failure.

Fixture generation (vendor license + sample end-user license for the Stage 2.2 simulated vendor delivery) happened in a separate holding directory on the supervisor side before the sub-agent was spawned; the fixture files were copied into the sub-agent's cwd at the Step 2.2 handoff moment per plan §3.4. No vendor tools were ever invoked inside the

sub-agent's cwd, so no vendor state files (.lck_cache_, .rg_gencount_) could leak into the isolated environment.

2.4 What this report covers / does not cover

COVERED:

- The three sub-agent passes, stage by stage.
- All 15 gaps surfaced (13 first-pass, 2 delta-pass), each with severity, description, in-session resolution (if any), and disposition.
- The two late-cycle defects (GA-002 and GA-013) that warranted their own entries in the QA Report's forensic late-cycle-defects section (now §4.2.7 and §4.2.8 there). Details reproduced here for self-containment.
- Process improvements banked for future Phase 4.1 cycles (§8).
- Zip SHAs and commit cross-references.

NOT COVERED:

- Plan §4.2 Tier enforcement and §4.3 Clock manipulation end-to-end, which are separate Phase 4 sub-phases covered by unit tests and the Phase 2B harness suite (see QA Report §3.4.2 and §3.4.3). Phase 4.1 Stage 5 does exercise Premium-tier floating-server seat caps but that is incidental cross-coverage, not the primary test of Plan §4.2.
- Phase 6 legal / branding review of the license agreement, which is a separate workstream tracked outside this report.

3. RUN 1: FIRST-PASS DRY-RUN

3.1 Setup

Date: 2026-04-24, first execution of the day. Operator: fresh general-purpose sub-agent via Claude Agent SDK (no worktree). Working directory: C:\tmp\phase4_1_onboarding\. Zip under test: rockyguard-v1.2.0-windows-x64-customer.zip, SHA-256 efe8284526fa65f103fce5d213d4fdbbfed54ea5f028a5a0563119e7fc29dfc2. Plan revision: RockyGuard_Phase4_1_Plan revision 3 (commit cd70644). Stage 0 pre-flight: PASS. No .git in cwd or any ancestor; git status fails; dev-tree files unreachable; only the extracted zip contents visible.

3.2 Stage-by-stage record

Stage	Step	Verdict	Notes
1	Cold read (3 docs)	PASS-with-gaps	Several undefined / ambiguous terms logged to gap list.
2	2.1 license_keygen	PASS	
2	2.2 vendor license handoff	FAIL-as-expected	Block documented (GA-001 HIGH); simulated delivery from fixture holding dir satisfied the block per plan §3.4.
2	2.3 license_create	PASS	Doc example missing --vendor-license (GA-003 MEDIUM).
2	2.4 license_verify	PASS	
2	2.5 rg_fingerprint	PASS-with-gap	Doc example output does not match actual (GA-005 MEDIUM).
3	3.1/3.2a first build attempt	FAIL (BLOCKER)	GA-002: linking per §3.1 fails with 15 unresolved externals (LNK2019 / LNK1120).
3	3.2b with workaround	PASS	After adding libssl.lib to the link line.
3	3.3 bundling + deploy + run	PASS	Deployment succeeded; my_app.exe printed "Licensed".
4	4.1 wrong key path	PASS	Error message clear.
4	4.2 remove .sig	PASS-with-gap	N/A for static build; docs could be more explicit (GA-006 LOW).
4	4.3 tamper DLL	N/A	Static build; no DLL in deployment.
4	4.4 past-date license	PASS	Clean "License has expired".
4	4.5 wrong-machine license	PASS	Clean diagnostic.
5	Premium (floating)	NOT TESTED	Time-boxed out; deferred to final-pass coverage.

3.3 Findings (13 gaps across all severities)

The first-pass sub-agent filed 13 gap entries. Full table reproduced in Appendix C for audit retention; a severity-grouped summary follows:

- 1 BLOCKER (release-blocker on onboarding): GA-002. Quick Start linking procedure in Customer_Documentation §3.1 omits libssl.lib and several required Windows system libraries. A fresh customer following §3.1 verbatim hits 15 unresolved externals.
- 2 HIGH (significant customer friction): GA-001 (vendor-license delivery channel undocumented; a customer who does not have vendor_license.json has no recourse in the docs), GA-013 (the shipped rockyguard.lib, libssl.lib, libcrypto.lib are all Release-CRT / MD builds; Debug-mode consumer builds hit LNK2038 / LNK1319 on _ITERATOR_DEBUG_LEVEL and RuntimeLibrary mismatches).

- 4 MEDIUM (docs inconsistency or omission that slows but does not block the customer): GA-003 (Quick Start Step 2 example omits `--vendor-license` flag), GA-004 (README.txt CONTENTS omits `deps/` directory where the bundled OpenSSL lives), GA-005 (`rg_fingerprint` example output in docs does not match actual tool output; component order and empty-component behavior undocumented), GA-009 (floating tools lack "Premium tier only" caveat in README).
- 6 LOW (cosmetic or forward-looking): GA-006 (static builds have no `.sig` file, docs never say so), GA-007 (C++17 requirement not explicitly stated anywhere), GA-008 (copyright line reads "[Legal Entity TBD]" which looks unfinished), GA-010 (date-only vs ISO-8601 datetime inconsistency between API Reference §8 and Customer_Documentation §4.3), GA-011 (API Reference §2 header list omits `export.h`), GA-012 (no explicit "what to ship to your end user" per-variant checklist).

3.4 In-session resolutions

Seven of the 13 gaps were fixed in-session by the supervising QA engineer as documentation patches. This was the right call for the BLOCKER (cannot ship a release that does not link) and for the HIGH CRT caveat (cannot ask customers to guess which CRT the library was built with), and for four MEDIUM items that were trivial text edits in the same files already being touched.

Resolved in-session (seven):

- GA-002 (BLOCKER): Customer_Documentation §3.1 and §3.2 rewritten with the full static link line: `rockyguard.lib + libssl.lib + libcrypto.lib + six Windows system libs (ws2_32, crypt32, iphlpapi, ole32, oleaut32, wbemuuid)`, plus the `deps/include` path. `examples/customer_examples_CMakeLists.txt` updated to include `crypt32` in the Windows link list. Re-verified by building a minimal consumer project against the shipped `rockyguard.lib` in Release mode: compiles, links, runs.
- GA-013 (HIGH, docs-only fix for v1.2.0): a prominent "IMPORTANT on Windows" CRT-variant caveat added to §3.1 explaining that the shipped library is `/MD` (Dynamic Release CRT) and the consumer must either build in Release mode or obtain a Debug-CRT build from the vendor. Medium-term fix (ship a matching Debug-CRT library alongside the Release one, or rebuild the shipped library with `/MT` static CRT so the consumer's CRT choice no longer has to match) is deferred to v1.2.1.
- GA-003 (MEDIUM): Customer_Documentation §2.3 Step 2 and the three §4.3 examples (node-locked, floating, permanent) now include `--vendor-license vendor_license.json`, with a preamble stating the tool will refuse to run without it.
- GA-004 (MEDIUM): `customer-use/README.txt` CONTENTS section now lists `deps/include/openssl/` and `deps/lib/libcrypto.lib + libssl.lib`, applied in the README heredocs of `build_and_package.sh` for both Windows and Linux zips.
- GA-005 (MEDIUM): Customer_Documentation §4.2 example output rewritten to match actual `rg_fingerprint -v` output, with a new note on the pipe-separated component order (MAC | CPU | Disk | Motherboard) and empty-slot representation.
- GA-009 (MEDIUM): `customer-use/README.txt` now lists `floating_server_example.exe` and `floating_client_example.exe` with the "(Premium tier only; requires Premium vendor license)" caveat.
- GA-007 (LOW, confirmation): verified that the prominent C++17 requirement block and `static_assert(__cplusplus >= 201703L, ...)` in the public header were already in place from the plan-authoring commit `4eb2fae`.

Deferred to v1.2.1 (five):

-
- GA-001 (HIGH, content-substantial): the vendor-license onboarding section is substantial new doc content (support-contact channel, expected format, sample filename, troubleshooting) and does not fit a one-session edit window.
 - GA-006 (LOW): static-build .sig clarification.
 - GA-010, GA-011, GA-012 (LOW, cosmetic).

Deferred to v1.3 (one):

- GA-008 (LOW): resolving the "[Legal Entity TBD]" copyright line is a business decision, not a technical fix. Tracked in Phase 6 (legal / branding review).

3.5 Verdict

Run 1 verdict: FAIL (one BLOCKER surfaced, seven findings closed in-session, six deferred).

The run accomplished what Phase 4.1 is designed to accomplish: it surfaced a release-blocker that the development team's familiarity with the library had hidden from an in-tree review. The BLOCKER fix was trivial once stated (~5-line doc edit plus one line in the examples CMakeLists), but would have been a support-ticket moment for every first customer without this run.

4. RUN 2: DELTA RE-RUN

4.1 Rationale

The in-session resolutions from Run 1 were applied by the supervising QA engineer, who by that point had seen the sub-agent's transcript and could no longer re-test the fresh-customer view. A second fresh sub-agent was therefore spawned to verify that:

1. The original four findings (GA-002, GA-003, GA-004, GA-005) no longer reproduce for a fresh reader.
2. The new CRT caveat (GA-013 fix) is visible and comprehensible in the shipped zip.
3. No regression was introduced by the doc edits (e.g. a broken cross-reference, a typo'd link-library name).

Delta scope: Stages 1, 2, 3 only (the areas where docs had changed). Stages 4 and 5 were unchanged from Run 1's documentation baseline and were explicitly skipped per the sub-agent's brief to avoid burning time re-running stable paths.

4.2 Setup

Date: 2026-04-24, approximately 3 hours after Run 1. Operator: fresh general-purpose sub-agent via Claude Agent SDK, no shared memory with the Run 1 operator. Working directory: C:\tmp\phase4_1_rerun\. Zip under test: rockyguard-v1.2.0-windows-x64-customer.zip regenerated after the Run 1 in-session fixes, SHA-256 2b0bb957f3426a967fdcf8e84a24951836bbf28a18a9136ff906ae9d9de574fc. Stage 0 pre-flight: PASS (same discipline as Run 1).

4.3 Resolutions confirmed

All four first-pass fixes tested by Run 2 are **RESOLVED** for a fresh reader:

First-pass gap	Delta-run evidence
GA-002 (BLOCKER) -- §3.1 missing libssl	§3.1 now lists rockyguard.lib + libssl.lib + libcrypto.lib + six Windows system libs + deps/include. Release-mode cmake --build against the shipped zip succeeded; my_app.exe printed "Licensed".
GA-003 (MEDIUM) -- Quick Start Step 2 missing --vendor-license	§2.3 Step 2 now includes --vendor-license vendor_license.json.
GA-004 (MEDIUM) -- README missing deps/	README CONTENTS now lists deps/include/openssl/ and deps/lib/libcrypto.lib + libssl.lib.
GA-005 (MEDIUM) -- rg_fingerprint docs mismatch	§4.2 example matches actual tool output; pipe-order and empty-slot note present.

4.4 New findings

Run 2 surfaced two new findings, both low severity:

DR2-001 (LOW, meta-process): the sub-agent could not find the CRT caveat from GA-013's fix in §3.1 of the unpacked zip. Investigation traced the cause to the fact that the zip had been regenerated by build_and_package.sh BEFORE the CRT caveat edit landed. Source docs contained the caveat; the shipped zip did not. This was fixed immediately by re-running bash build_and_package.sh to re-synchronize the zip with the current source tree, and verified post-rebuild: grep -c 'IMPORTANT on Windows' customer-use/docs/Custom Documentatation.txt returns 1. The new zip SHA was recorded in archives/MANIFEST.txt.

DR2-002 (LOW, already tracked): §2.2 still does not give a vendor contact channel for a customer who does not have vendor_license.json. Same as first-pass GA-001, already deferred to v1.2.1. Re-confirmed in this delta run; no new action.

4.5 Verdict

Run 2 verdict: PASS with doc patches (both new findings are low severity; DR2-001 is closed, DR2-002 is already tracked for v1.2.1 as GA-001).

The delta re-run was worth running. It caught DR2-001, which the Run 1 post-fix verification did not: that verification had built a consumer project using the UPDATED SOURCE docs text, which included the CRT caveat in the source tree but had no way to reveal that the shipped zip was out of date. A fresh sub-agent unpacking the zip saw what a real customer would see, and the staleness was immediately apparent. Process improvement banked for future Phase 4.1 cycles (§8).

5. RUN 3: FINAL FULL RE-RUN

5.1 Rationale

Runs 1 and 2 together left two open questions:

1. After the DR2-001 zip-regeneration, does a full Stage 0-4 walkthrough still pass with no new findings? 2. Does the Premium-tier Stage 5 (floating_server_example + floating_client_example) work end-to-end as documented? Stage 5 was time-boxed out of Run 1 and explicitly excluded from Run 2's delta scope, so neither prior pass had exercised it.

A final clean full-scope pass was therefore commissioned with a third fresh sub-agent.

5.2 Setup

Date: 2026-04-24, final execution of the day (post-DR2-001 zip rebuild). Operator: fresh general-purpose sub-agent via Claude Agent SDK, no shared memory with the Run 1 or Run 2 operators. Working directory: C:\tmp\phase4_1_final\
 Zip under test: rockyguard-v1.2.0-windows-x64-customer.zip, SHA-256
 6f1634be700282c30667f4e74626276ca554045b665fa73d517d7855b8684a49 (pinned in archives/MANIFEST.txt).
 Stage 0 pre-flight: PASS.

5.3 Stages 0 through 5: all PASS

Final-run verdict per stage:

Stage	Scope	Verdict	Notes
0	Pristine environment setup	PASS	No .git, no dev-tree visibility.
1	Cold read (3 docs)	PASS	All prior gap-list items confirmed addressed; no new readability issues.
2	Quick Start (five steps)	PASS	2.1 keygen, 2.2 simulated vendor delivery per plan §3.4, 2.3 license_create with --vendor-license, 2.4 license_verify, 2.5 rg_fingerprint. All clean.
3	Minimal library integration	PASS	§3.1 link line from the fixed docs builds cleanly in Release mode. Bundling and deployment succeed.
4	Adversarial probes	PASS	Wrong key path, stripped .sig (static-build note now present), tampered DLL (N/A for static), past-date license, wrong-machine license -- all fail loudly and clearly as the plan requires.
5	Premium (floating)	PASS	Floating server and client built from examples/floating_*_example.cpp. Handshake succeeded; checkout / heartbeat / release cycle completed; server correctly enforced seat cap on a second client. First end-to-end Stage 5 execution in the v1.2.0 cycle.

All five Phase 4.1 gating questions per Plan §6 answered affirmatively:

- Q1. Can a fresh customer build the Quick Start integration from the shipped docs alone? Yes.
- Q2. Do the tools produce the outputs the docs promise? Yes.
- Q3. Do adversarial probes (wrong key, stripped .sig, tampered DLL, past-date, wrong-machine) fail loudly and clearly? Yes.
- Q4. Does the Premium-tier floating server / client path work end-to-end from the examples as documented? Yes.
- Q5. Are there any BLOCKER findings remaining? No. (One HIGH, GA-001, is deferred to v1.2.1 per §6.2 -- an acceptable pass condition under the QA Plan.)

Zero new findings at any severity. All 15 prior findings (GA-001 through GA-013 plus DR2-001 and DR2-002) confirmed either RESOLVED in-zip or correctly DEFERRED as already tracked.

5.4 Verdict

Run 3 verdict: PASS -- ready to sign off for v1.2.0.

The meta-process improvement from Run 2 (rebuild the zip after every in-session doc fix, verify against the zip) held: this final run saw a zip that matched the current source tree exactly, and the sub-agent raised no staleness findings.

6. CONSOLIDATED GAP LIST AND RESOLUTIONS

6.1 Resolved in v1.2.0 (in-zip at tag)

Gap ID	Severity	One-line description	Resolution commit / artifact
GA-002	BLOCKER	Quick Start link line missing libssl + Windows syslibs	Customer_Documentati on.txt §3.1/§3.2 rewrite; examples CMake; re-verified in Runs 2 and 3
GA-013	HIGH	Release-CRT caveat missing (Debug consumers hit LNK2038)	Customer_Documentati on.txt §3.1 "IMPORTANT on Windows" block; medium-term Debug-CRT library ship deferred to v1.2.1
GA-003	MEDIUM	Quick Start Step 2 missing --vendor-license flag	Customer_Documentati on.txt §2.3 and §4.3 (three examples)
GA-004	MEDIUM	README CONTENTS missing deps/entry	build_and_package.sh README heredocs (Windows + Linux)
GA-005	MEDIUM	rg_fingerprint docs example does not match actual output	Customer_Documentati on.txt §4.2 rewrite with pipe-order note
GA-009	MEDIUM	README omits Premium-tier caveat on floating binaries	build_and_package.sh README heredocs
GA-007	LOW	C++17 requirement not stated	Verified already present (static_assert + docs block from commit 4eb2fae)
DR2-001	LOW	Shipped zip predated the CRT caveat fix	bash build_and_package.sh re-run; new zip SHA pinned in archives/MANIFEST.txt

6.2 Deferred to v1.2.1

Gap ID	Severity	One-line description	Rationale for deferral
GA-001	HIGH	Vendor-license delivery channel / contact undocumented	Substantial new doc content; scoping a support-channel paragraph is a product-side decision.
GA-006	LOW	Static-build .sig note asymmetric	Clarity improvement; not a regression.
GA-010	LOW	Date-only vs ISO-8601 datetime inconsistency	Cosmetic; requires agreement on canonical form.
GA-011	LOW	API Reference §2 omits export.h	Cosmetic.
GA-012	LOW	No per-variant "what to ship to end user" checklist	Additive; does not contradict existing guidance.
GA-013 (medium-term)	HIGH	Ship a matching Debug-CRT library alongside Release	Build-system work: add lib/static/rockyguard_mdd.lib + deps/lib/libssl_mdd.lib + libcrypto_mdd.lib, with CMake Release/Debug-detecting imported target. Docs caveat stabilises v1.2.0.

6.3 Deferred to v1.3

Gap ID	Severity	One-line description	Rationale for deferral
GA-008	LOW	Copyright line reads "[Legal Entity TBD]"	Business decision, not a technical fix. Tracked in Phase 6.
(new)	--	Ship rockyguard-config.cmake imported target	Would make GA-002-class defects impossible by construction; non-trivial build-system work.

Gap ID	Severity	One-line description	Rationale for deferral
(new)	--	CI job replicating this dry-run's consumer build	Prevents GA-002-class regressions in future releases.

7. LATE-CYCLE DEFECTS REGISTERED IN THE QA REPORT

The following two defects, though surfaced by a documentation-audit exercise rather than a code-test harness, are proper late-cycle defects because (a) they would have been customer-impactful had they reached the field and (b) neither was previously tracked in the numbered issue list. Both are recorded in RockyGuard_QA_Report.txt §4.2.7 and §4.2.8 respectively; summaries follow for this report's self-containment.

7.1 GA-002 (BLOCKER): Quick Start linking procedure incomplete

Field	Detail
Severity	BLOCKER (release-blocker on customer onboarding)
Platform	Windows and Linux (documentation-only; same link line shipped on both)
Present in	Pre-release docs from early Phase 4 through rc2. Never reached any customer -- v1.2.0 is the inaugural shipping release.
Fixed in	1.2.0
Validated	Phase 4.1 sub-agent delta re-run and final re-run: a fresh customer following §3.1 verbatim in Release mode builds the consumer project cleanly and links against the shipped rockyguard.lib.

Description: Customer_Documentation §3.1 "Linking (Static)" told the customer to link only OpenSSL::Crypto. The shipped static rockyguard.lib in fact has unresolved references to libssl (SSL_CTX_new, TLS_client_method, and others) introduced by the time-anchor HTTPS online-verification path (fix #4.2.3 from the Phase 2 late-cycle-defect list). A fresh customer following §3.1 hits LNK2019 x15 and LNK1120: 15 unresolved externals. The examples CMakeLists had the same omission. The required library (libssl.lib) IS shipped under customer-use/deps/lib/ -- the docs just did not tell the customer to link it.

Fix: rewrote Customer_Documentation §3.1 and §3.2 with the full link line (rockyguard.lib + libssl.lib + libcrypto.lib + ws2_32 + crypt32 + iphlpapi + ole32 + oleaut32 + wbemuuid) and the deps/include path. Added crypt32 to examples/customer_examples_CMakeLists.txt (the other five Windows system libs were already present). Re-verified by Runs 2 and 3 sub-agents building minimal consumer projects against the shipped zip in Release mode: each linked cleanly and produced a working my_app.exe.

7.2 GA-013 (HIGH): Release-CRT library cannot serve Debug consumers

Field	Detail
Severity	HIGH (blocks Debug-mode development against the library)
Platform	Windows (MSVC); no equivalent on Linux because libstdc++ has a single ABI across build types.
Present in	Pre-release library builds from early development through rc2. Never reached any customer -- v1.2.0 is the inaugural shipping release.
Fixed in	1.2.0 (documentation fix; medium-term library-build fix deferred to v1.2.1)
Validated	Phase 4.1 sub-agent final re-run: the CRT caveat is present in shipped §3.1 and reads accurately against the shipped artifacts.

Description: the shipped rockyguard.lib and the bundled deps/lib/libssl.lib / libcrypto.lib are built with the Release CRT (/MD). A customer attempting to build a consumer application in Debug mode hits LNK2038 on _ITERATOR_DEBUG_LEVEL (0 vs 2) and RuntimeLibrary (MD_DynamicRelease vs MDd_DynamicDebug), plus LNK1319 with 1 mismatch detected. The customer cannot develop against the library in Debug mode without a matching Debug-CRT library build.

Fix (short-term, v1.2.0): added a prominent "IMPORTANT on Windows" CRT-variant caveat to

Customer_Documentation §3.1 explaining that the shipped library is /MD (Dynamic-Release) and the consumer project must build in Release mode. Fix (medium-term, v1.2.1): ship a matching Debug-CRT library build (lib/static/rockyguard_mdd.lib + deps/lib/libssl_mdd.lib + deps/lib/libcrypto_mdd.lib) with a Release/Debug-detecting CMake imported target. Alternative under evaluation: rebuild the shipped library with /MT (static CRT) so the consumer's CRT choice no longer has to match.

8. META-FINDINGS AND PROCESS IMPROVEMENTS

Phase 4.1's three-pass structure surfaced one meta-finding worth banking for future Phase 4.1 cycles:

Always re-run `bash build_and_package.sh` AFTER every in-session doc fix, and verify against the packaged zip -- NOT against the source tree.

Rationale, reconstructed from DR2-001: the Run 1 post-fix verification built a consumer project using the updated source docs text. That verification succeeded on the local machine because the local file system had the edit. But the customer-use zip (which is what a real customer actually receives) had been built by `build_and_package.sh` BEFORE the last doc edit landed. The shipped zip was one fix out of date. The Run 1 verification had no way to reveal the gap because it wasn't reading from the zip. A fresh sub-agent in Run 2, unpacking the zip in an isolated temp directory, saw what a real customer would see -- and immediately filed DR2-001.

Two smaller process improvements also banked:

1. Fixture files for the Stage 2.2 simulated vendor delivery must be generated OUTSIDE the sub-agent's cwd and copied in at the handoff moment. The first fixture generation script (`qa/onboarding/samples/generate_samples.sh`) ran with `cwd=temp` and leaked `.lck_cache_vendor` and `.rg_gencount_*` into the sub-agent's environment, which is a real-customer-can-see-these-files leak. Fixed in the script: vendor tool invocations moved to a dedicated holding directory.
2. Don't combine `set -o pipefail` with `grep | head -1`. A SIGPIPE from head closing early propagates back as a failure under pipefail. Replace with `awk '/pattern/ {print; exit}'`. Applied to `qa/onboarding/samples/generate_samples.sh`.

9. RISK ASSESSMENT AND SIGN-OFF

Overall residual risk after Phase 4.1: LOW.

Risk class	Likelihood	Mitigation
A new customer fails to link the library	Very low	GA-002 closed in v1.2.0; verified by two independent fresh sub-agents against the shipped zip.
A new customer hits Debug-CRT mismatch	Low	GA-013 short-term docs fix closed in v1.2.0; medium-term library-build fix deferred to v1.2.1. Debug-build customers are explicitly told to build in Release mode or request a Debug-CRT build.
A new customer has no vendor_license.json and does not know whom to contact	Low	GA-001 (vendor-license onboarding content) deferred to v1.2.1; any v1.2.0 customer reaches this through sales channel, not by reading the docs cold.
Regression on onboarding in a future release	Low	Process improvement banked (§8); a CI job that replicates this dry-run's consumer build is tracked for v1.3.
Stage 5 Premium path breaks	Low	First end-to-end validation in this cycle was Run 3; path is covered.

Phase 4.1 sign-off requested from:

Role	Scope of sign-off
Technical lead	Acceptance of the 15-gap disposition (7 resolved in-zip for v1.2.0; 6 tracked for v1.2.1; 3 for v1.3).
QA lead	Acceptance of the three-pass sub-agent methodology as the Phase 4.1 execution record.
Product owner	Acceptance of GA-001, GA-010, GA-011, GA-012 deferral to v1.2.1 and GA-008 deferral to Phase 6.

With the sign-offs above, Phase 4.1 closes. The only remaining pre-tag workstream for v1.2.0 is Phase 6 (legal / branding review of the license agreement and copyright line).

Appendix A: ARTIFACT SHA-256s (CUSTOMER-USE ZIP, PINNED PER RUN)

Run	Zip	SHA-256
1	rockyguard-v1.2.0- windows-x64-customer.zip (pre-fix)	efe8284526fa65f103fc e5d213d4fdbbfed54ea5 f028a5a0563119e7fc29dfc2
2	rockyguard-v1.2.0- windows-x64-customer.zip (post-Run-1 fix)	2b0bb957f3426a967fdc fae84a24951836bbf28a 18a9136ff906ae9d9de574fc
3	rockyguard-v1.2.0- windows-x64-customer.zip (post-DR2-001 rebuild; shipped)	6f1634be700282c30667 f4e74626276ca554045b 665fa73d517d7855b8684a49

The complete history of every zip this repository has produced is kept in archives/MANIFEST.txt. The v1.2.0 tag will point at the Run 3 SHA (pending any further Phase 6 doc-only edits that trigger another rebuild).

Appendix B: SOURCE-COMMIT CROSS-REFERENCE

Workstream	Representative commit(s)
Phase 4.1 Plan (initial draft through revision 3)	8d70fe8, 52aedd1, c8cce8c, a4ad146, 59b06a4, 4eb2fae, 071e77f, ecefc18, b13072a, a4819b2, bc91e8b, cd70644
Run 1 execution + in-session fixes	ef5de0d
Run 2 delta re-run + DR2-001 zip rebuild	49be9b0
Run 3 final full re-run + QA Report Phase 4.1 row	7e0aa0c
QA Report restructure (Phase 4 as real-world scenario)	33bf989, 1077a79, c76c336
QA Report stale-content sweep (post-Phase-4.1 verdict)	845da74
Phase 4.1 Report (this document)	(this commit)

For the complete history see `git log --oneline --grep="Phase 4.1"`.

Appendix C: FULL GAP-LIST TABLE (FIRST PASS)

Reproduced verbatim from qa/onboarding/RESULTS.md for audit retention.

Gap ID	Stage / Step	Severity	Description	Suggested fix
GA-001	2 / 2.2	HIGH	§2.2 says the customer "should have received" a vendor_license.json but never explains delivery channel, format, what to do if missing, how to request one, or tier differences. A fresh customer has no idea what to look for or whom to contact.	Add §2.2.1 "How you receive your vendor license" with sales/support contact, expected delivery format, sample filename, troubleshooting, and what's inside the file.
GA-002	3 / 3.1	BLOCKER	§3.1 "Linking (Static)" tells the customer to link only OpenSSL::Crypto. Actual static rockyguard.lib has unresolved SSL_* / TLS_client_method symbols from time_anchor.obj. Build fails with LNK2019 x15 and LNK1120.	Update §3.1 to list both libssl and libcrypto plus Windows system libs. Apply the same fix to examples CMakeLists. Long-term: ship rockyguard-config.cmake.
GA-003	2 / 2.3	MEDIUM	§2.3 Step 2 shows a license_create command that omits --vendor-license. Tool's --help and API §8 both say it's required.	Add --vendor-license vendor_license.json \ to the Step 2 example. Mirror in README QUICK START.
GA-004	1 / README.txt	MEDIUM	README CONTENTS omits deps/ directory.	Add deps/ entry listing deps/include/openssl/ and deps/lib/libcrypto.lib + libssl.lib.
GA-005	2 / 2.5	MEDIUM	§4.2 shows example rg_fingerprint -v output with a "Fingerprint:" label line, but actual output prints the hash on an unlabeled line. Docs never state component order in the pipe-separated hash, nor behavior for missing components.	Replace the docs example with literal actual output. Add a note under §6.1 stating the fixed order (MAC)
GA-006	4 / 4.2	LOW	Docs say ship rockyguard.sig next to the DLL (§9.3) and the README LINKING/Shared block lists it. No symmetric explicit note in STATIC section saying "static builds do NOT need rockyguard.sig".	Add one sentence in §3.1 (Static): "Static builds have no separate integrity file." Add reciprocal sentence in §9.3.
GA-007	3 / 3.2a	LOW	Docs never explicitly require C++17. In this run MSVC's default was high enough so no error observed, but a customer on a strict toolchain would hit the library's static_assert.	Add set(CMAKE_CXX_STANDARD 17) / set(CMAKE_CXX_STANDARD_REQUIRED ON) to §3.1 / §3.2. (Pre-emptively added during plan drafting; verified in Run 2.)
GA-008	1 / all docs	LOW	Copyright reads "Copyright (c) 2025-2026 [Legal Entity TBD]". Shipping with TBD legal entity looks unfinished.	Replace with actual legal entity, or omit until set. Affects Customer_Documentation, Customer_API_Reference, License Agreement.
GA-009	1 / README.txt	LOW	README lists floating binaries under tools/ without the "(Premium tier only)" caveat.	Add "(Premium tier only; requires Premium vendor license)" beside the two floating binaries.
GA-010	1 / API Reference	LOW	§8 example shows --expires "2027-12-31" (date only) but §4.3 of main doc uses full ISO 8601. Unclear whether date-only is accepted.	Pick one canonical form; document tolerance explicitly.

Gap ID	Stage / Step	Severity	Description	Suggested fix
GA-011	1 / API Reference	LOW	§2 "Namespace and Headers" omits export.h which README lists.	Add export.h to the API Reference header list or note it's internal-only.
GA-012	3 / 3.3	LOW	No "what to ship to your end user" per-variant checklist.	Add a short "What to ship to your end user" section with a per-variant checklist.
GA-013	3 / 3.2 verif.	HIGH	Shipped rockyguard.lib and bundled libssl.lib / libcrypto.lib are Release-CRT (/MD). Debug consumer builds hit LNK2038 / LNK1319 on _ITERATOR_DEBUG_LEVEL and RuntimeLibrary.	Short-term: CRT-variant caveat in §3.1. Medium-term: ship matching Debug-CRT library. Alternative: rebuild with /MT.

End of report.