# L3dgr SEC 17a-4 and WORM Readiness Checklist

Document ID: L3DGR-SEC17A4-2.3
Version: 2.3
First published: May 20, 2026
Effective date: August 2, 2026
Last updated: September 11, 2026
Document type: Compliance review worksheet (not a legal policy)

## IMPORTANT NOTICE

This checklist is a technical review aid for operators, compliance teams, and auditors. It is not legal advice, not a regulatory filing, and not a certification that your organization satisfies SEC Rule 17a-4(f), FINRA requirements, or any other rule. Fors33, Inc. ("Fors33") does not guarantee compliance outcomes. You and your counsel remain responsible for retention programs, supervisory procedures, timestamp authority selection, storage policies, and evidentiary use in examinations or litigation.

Fors33 L3dgr provides cryptographic integrity tooling (scan, seal, verify, audit export, and optional remote sync). It helps you produce verifiable evidence on infrastructure you control. It does not replace WORM storage you approve, legal hold process, or examiner-specific interpretations.

How to use this worksheet

Work through each phase with your records environment in mind. Mark items you can demonstrate today, items that require process change, and items that require infrastructure you have not yet deployed. Keep evidence (policies, configuration screenshots, sample manifests, audit exports) with your review packet.

Regulatory context (summary)

SEC Rule 17a-4(f) and related FINRA expectations require firms to preserve certain electronic records in a non-rewriteable, non-erasable format (often summarized as WORM), with audit trails and timely production. Ordinary mutable backups, unlocked object storage, or editable working copies without independent integrity proof may not meet your obligations. The questions below focus on technical controls that support defensible chain of custody, not on substituting this PDF for counsel review.

### Phase 1: Ingestion and deterministic integrity

`[ ] Point-of-origin integrity: Are records hashed or sealed at or before the point you treat them as authoritative, rather than only after mutable copies exist?`

[ ] Algorithm policy: Do you standardize on recognized hash or signature algorithms (for example SHA-256 or BLAKE3) and document allowed algorithms for your program?

[ ] Time association: Can you associate records with verifiable time evidence (system time plus, where required, independent timestamping) at seal or ingest?

[ ] Change detection: Can you detect post-ingest modification to sealed or referenced files without re-reading entire corpora manually?

### Phase 2: Storage immutability (WORM and write policy)

[ ] Non-rewriteable target: For records in scope, is there a designated WORM or object-lock destination (or equivalent control) that prevents overwrite and deletion for the retention period?

[ ] Separation of duties: Are administrators who can delete production data distinct from roles that approve retention policy, where your program requires it?

[ ] Proof of immutability: Can you demonstrate to an auditor how your storage layer enforces immutability (bucket policy, legal hold, vault lock, or vendor attestation), not only application claims?

[ ] Copy vs original: Can you identify which object is the authoritative record and which copies are derivatives?

### Phase 3: Cryptographic chain of custody

[ ] Sidecar or manifest discipline: Does each production dataset carry an integrity artifact (for example a .f33 sidecar, manifest, or receipt) that binds file identity to metadata?

[ ] Independent verification: Can integrity be verified locally with open or customer-controlled tools without sending raw files to a vendor cloud?

[ ] Drift monitoring: Do you have a defined process to re-verify integrity on a schedule or before downstream use (ML training, e-discovery export, model release)?

[ ] Manifest authenticity: Where manifests are used, are they signed or HMAC-bound so off-host tampering is detectable?

### Phase 4: Audit logging and least-privilege access

[ ] Append-only audit trail: Is there an append-only log of security-relevant and operator actions (access, seal, verify, configuration changes) with retention aligned to your policy?

[ ] Local-first processing option: Can scanning and verification run without exporting raw proprietary files to the public internet when your policy requires it?

[ ] Operator attribution: Are actions attributable to named operators or service accounts, not only shared machine credentials?

[ ] Connector and egress control: If data leaves the environment (connectors, BYOB upload), is that path documented, approved, and monitored?

### Phase 5: Retention, legal hold, and production

[ ] Retention schedule: Are retention periods defined per record class and enforced without relying on informal administrator memory?

[ ] Legal hold: Can you suspend deletion for subsets under hold without breaking integrity evidence for unrelated records?

[ ] Production format: Under examination or subpoena, can you produce records together with integrity proof and audit context in formats your counsel approves?

[ ] Uninstall and lifecycle: If desktop or extension tooling holds keys or audit state locally, is export and backup required before decommission (see L3dgr Privacy Policy L3DGR-PRIVACY-2.11)?

Liability assessment

Unchecked items indicate gaps in your technical or procedural chain of custody. Gaps may be acceptable only where your counsel and compliance officers agree a control is out of scope for a given dataset class. Do not treat this worksheet as an automatic pass/fail certification.

How Fors33 L3dgr relates to this checklist (informational)

L3dgr is designed to help operators working on Docker-accessible paths and local volumes:

- Scan and compare integrity; generate .f33 sidecars and manifests on paths you select.
- Seal and verify with receipts suitable for audit export (including Auditor Package workflows in regulated configurations).

- Maintain append-only style audit events in extension-managed state for many operator actions.
- Run verification using public fors33-scanner and fors33-verifier images where appropriate.
- Optional operator-configured remote storage and RFC 3161 timestamp services when you enable them; you choose the TSA and bucket policies your program requires.

L3dgr does not, by itself: select your WORM vendor, configure bucket object locks, satisfy all FINRA books-and-records obligations, or certify your firm. Map product features to your control environment with internal compliance and counsel.

L3dgr supplies integrity evidence on customer-controlled storage. It is not a 17a-4 electronic recordkeeping system. Fors33 does not file a Rule 17a-4(i) third-party undertaking. The 2022/2023 audit-trail alternative to WORM is the firm's storage choice, not this product.

Suggested next steps

1. Complete this worksheet with your team and retain the dated PDF in your review file.
2. Review L3dgr legal and privacy documents at fors33.com/l3dgr/docs (L3DGR-EULA-2.7, L3DGR-PRIVACY-2.11).
3. Evaluate L3dgr on representative volumes in a non-production environment before regulated use.
4. Request the gated checklist download or support routing at fors33.com/l3dgr/audit when you need a formal copy for procurement records.
5. Contact fors33.com/l3dgr/support for product questions; contact legal@fors33.com for policy questions.

Public integrity tools (verify independently)

- fors33/fors33-scanner and fors33/fors33-verifier on Docker Hub for baseline scan and verify workflows documented at fors33.com/l3dgr.

Document control

Supersedes informal or draft checklist versions without a Document ID. Revisions will increment the version number and Last updated date on fors33.com/l3dgr/docs. The Effective date stays the incorporation date unless Fors33 sets a new in-force date.