

FHIRBridge Interface Engine Submission Guide

Operational guide for moving source-system intake through FHIR generation, validation, queueing, mock/dry-run transport, live transport readiness, monitoring, and replay controls.

Submission Stages

1. Define source intake contract

Purpose: Make every inbound source row predictable before it enters conversion or queue processing.

Action: Document accepted formats, required identifiers, date formats, tenant context, source-system name, and the canonical row fields expected by Live Feed.

Readiness gate: Every source interface has a named owner, sample payload, field contract, and rejection rule.

Evidence: Source intake contract, sample payload, owner list, and rejected-row rules.

2. Validate inbound payload

Purpose: Stop incomplete or unsafe records before Bundle generation.

Action: Run preflight checks for tenant, organization/site/program identifiers, client identifiers, mandatory PDS fields, dates, and local code values.

Readiness gate: Invalid rows produce clear diagnostics and are not queued for submission.

Evidence: Preflight output, invalid-row diagnostics, and remediation owner.

3. Generate FHIR Bundle

Purpose: Convert canonical intake into traceable FHIR R4 content.

Action: Generate Patient, Organization, Location, EpisodeOfCare, Encounter, Appointment, Observation, Condition, ServiceRequest, and related resources as applicable.

Readiness gate: Each Bundle has stable identifiers, source traceability, and the expected resource count for the source row.

Evidence: Bundle JSON, conversion summary, resource inventory, and source-row trace.

4. Run validation before queueing

Purpose: Prevent known-bad Bundles from moving into transport.

Action: Run local requirement validation, terminology checks, FHIR structural validation, and authorized package validation where configured.

Readiness gate: No unresolved fatal/error validation issue exists before a job is eligible for submission.

Evidence: Validation report, OperationOutcome diagnostics, and issue disposition.

5. Queue submission job

Purpose: Create an auditable unit of work for retry, support, and operations review.

Action: Queue the validated Bundle with tenant, mode, endpoint target, status, attempt count, Bundle ID, and validation summary.

Readiness gate: Queued jobs include enough metadata to support retry and audit without opening raw PHI unnecessarily.

Evidence: Feed job record, queue status, Bundle ID, and validation summary.

6. Process mock or dry-run transport

Purpose: Prove the engine workflow without sending live production data.

Action: Process the next job in mock or dry-run mode, capture response code, response body, remote/mock ID, and OperationOutcome.

Readiness gate: Dry-run jobs reach accepted/mock-accepted or fail with actionable diagnostics.

Evidence: Attempt history, mock/dry-run response, and job status transition.

7. Configure live transport controls

Purpose: Prepare endpoint and credential handling for approved live submission.

Action: Configure endpoint URL, token ownership, secret rotation, TLS expectations, timeout, retry policy, and support contacts.

Readiness gate: Live mode remains disabled until onboarding approval and go/no-go sign-off are complete.

Evidence: Transport configuration checklist, credential owner, rotation note, and go/no-go record.

8. Monitor queue operations

Purpose: Keep production submissions observable and supportable.

Action: Review queued, validating, submitting, accepted, failed, and retrying jobs; investigate stale jobs and repeated failures.

Readiness gate: Operations has a daily review process, escalation path, and retained job evidence.

Evidence: Daily queue review, incident log, retry history, and support handoff notes.

9. Manage failures and replay

Purpose: Recover safely from validation, endpoint, credential, or remote response failures.

Action: Classify failure root cause, correct the source or configuration, regenerate/validate when needed, and submit a new traceable job.

Readiness gate: Replays are deliberate, traceable, and do not silently duplicate accepted submissions.

Evidence: Failure analysis, corrected payload, replacement job ID, and duplicate-prevention note.