High-Level Design Document: Octane Integration Overview

# Objective

This document provides a high-level overview of the integration between QzRelease2 and Micro Focus ALM Octane. It describes the main workflow and outlines the API endpoints used along with their access frequency.

# Integration Workflow Summary

1. User triggers test upload from QzRelease2 using a regression Jira ID.  
2. QzRelease2 generates and formats test results into Octane-compatible XML.  
3. Octane workspace and credentials are identified using a pre-mapped Octane ID.  
4. Test results are uploaded in batches to Octane using the test-results API.  
5. The status of each task is monitored and retried up to a threshold limit.  
6. After successful completion, Jira issues are linked to the respective test runs.  
7. Application modules and test suites are created dynamically if not present.

# Octane API Usage Summary

|  |  |  |  |
| --- | --- | --- | --- |
| API Endpoint | Purpose | Method | Frequency |
| /test-results | Upload test results | POST | 1 per 1000 tests |
| /test-results/{task\_id} | Check task status | GET | Every 5s, max 5 times |
| /test\_suites | Create or fetch test suite | GET / POST | Once per Jira |
| /application\_modules | Create or fetch application module | GET / POST | Once per path level |
| /stories?query=... | Resolve Jira to Octane ID | GET | Once per Jira |
| /automated\_tests | Retrieve test run data | GET | Once per upload verification |
| /automated\_tests/{id} | Link backlog coverage | PUT | Once per test |
| /suite\_runs | Fetch test suite run summary | GET | Once post upload |
| /shared\_spaces/{space\_id}/workspaces/{workspace\_id}/application\_modules?query=name EQ | Fetch module ID by name | GET | One-time during config setup |

# Configuration & Retry Strategy

Configuration is initialized using a structured Octane ID string in the format: `spaceId-workspaceId-\*-octaneId`. Based on this, the following parameters are derived:

- space\_id and workspace\_id for API routing

- credentialsPath for authentication using secure vault

- rootUrl for workspace-level API communication

The root application module ID is fetched dynamically if not present, using the application module name with a GET request to the `/application\_modules` endpoint.

# Batching Strategy

Test results are uploaded in batches of 1000 per request to the `/test-results` endpoint. The slice size is configurable and can be adjusted in the application settings.

# Retry Mechanism

Each task returned from the upload operation is polled every 5 seconds using the `/test-results/{task\_id}` endpoint. A maximum of 5 retries is performed per task.

If the task is still in 'RUNNING' or 'QUEUED' state after 5 retries, it is added to a pending task tracker. These tasks are monitored separately, and if they remain unresponsive for over 10 minutes, the operation is marked as failed and logged for investigation.