Automation Tool Interface (ATI) Enterprise Guideline

1. Scope

This Enterprise Technology Guideline (ETG) represents the technical architecture standard for using an Automation Tool Interface (ATI) to address reoccurring software development and testing problems in the Office of Information and Technology (OIT). This guideline also refers to engineering guidelines for automated testing that can be reused by product teams shipping software solutions to Veterans and VA staff.

2. Background

Product teams face numerous challenges finding ways to get test automation tools up and running in a repeatable way across product lines in the Product Delivery Service (PDS). There is a need to have seamless access to tools enterprise-wide that can be used in different portfolios. Product line managers need to know where they can identify and select the appropriate tools depending on the business need.

3. Solution

VA is required to maintain an enterprise capability for Software Testing Operations (QAOps) across the product lines. QAOps helps product lines implement and maintain automated software testing solutions, by incorporating an Automation Tool Interface (ATI) into their development activities. The ATI combines free open-source software tools currently used by OIT, with a library of functions (APIs), to create a framework that adapts to range of testing requirements and solution platforms such as Web, VistA, Windows, Section 508, and application programming interface (API) applications. QAOps provides training and support for this proven framework which is currently incorporated in multiple Continuous Integration/Continuous Delivery (CI/CD) pipeline solutions.

Onboarding with the ATI is included for internal users at the Catalog of Developer Essentials (CODE VA) at https://code.va.gov/catalog/default/component/ati. Please visit the SharePoint site located in the References section for more information on how to use the ATI and to contact representatives in QAOps.

4. Technical Details

Highlights

- Developed using Java / Eclipse or other integrated development environment (IDE).
- Designed around a 3-tier architecture.
  - Automation Cores (Platform Independent Cores (Automation Tool Interface))
  - Page / Form Definitions / Interfaces (Application Specific)
  - Test Scripts / Suites (Application Specific)
- Standardized coding practices, test data, and execution result logging
- Jira Xray repositories house tests cases, scripts, and execution results
- Supports various test script writing tools; Junit, TestNG, Cucumber, and behavior-driven design (BDD)

Tools/Technologies

- Java / Eclipse
- Java JAX-RS and JAXB (Middle Tier / Web Services testing support)
- Java-A11y API (Section 508 scans)
- Selenium (Automation Tool Interface; Web applications only)
- WinAppDriver (Appium) (Automation Tool Interface; Windows applications only)
- Java SSH3 Connection (VistA application)
- Maven (Dependency Management)
- GitHub Package Manager
- JIRA Xray (Test Management /Reporting)
- Test Persona Manager – Single Sign-on (SSO)
- JMeter – performance testing
- Database – Java SQL

How to Connect to MTS

With ATI automation, product lines have options for integrating with their DevOps.

- MTS has created a store front for MTS supported tools. To get started visit the README document.
- ATI has created a va-ati-tenant branch in the MTS store front for ATI builds and ATI testing using ATI to test itself. Product lines can leverage this work by having

their own pipelines inside the va-ati-tenant Jenkins. Reach out to QAOps for how to leverage this work and add a pipeline for your project inside the va-ati-tenant branch. See the "References" section for link to QAOps SharePoint.

- Create your own! ATI based automation tests can integrate into most pipelines be they in Azure, AWS, AITC etc.

ATI provides guides for implementing ATI based automation tests. From workstation setup, project creation and structure, naming standards, coding standards, branching strategies, etc. Pulling all this together is our comprehensive guide for conducting peer reviews/Pull Request reviews. From a QAOps perspective, the more the code adheres to the review guidelines, the more acceptable the ATI implementation.

5. Evaluation for Incoming Applications

- Evaluate candidate application for Test Automation feasibility.
  - Identify the application development technology / platform (e.g., Java Web, Delphi, .NET)
  - Evaluate candidate application design.
  - Review against existing automation Core platform support
- Application platforms we currently support.
  - Web, Windows, VistA applications, and APIs
- Test Case / Script development.
  - Mentoring Phase 1 – Page class development approximately 4 weeks
  - Mentoring Phase 2 – Test Script development approximately 4 weeks
  - MTS Team Hosting Environment
- CI / CD Pipeline
  - Jenkins
  - Jira/Xray

6. Complementary Tools and Patterns

VistA Automated Testing Suite (VATS)

- JRE-based suite for testing VistA systems, usable for unit, component, functional, and regression across MUMPS and web interfaces within VistA.

Cypress (for VA.gov front-end)

- Best practices documented in internal VA GitHub repo include:
    - Use of custom commands (e.g., cy.login), fixture data, viewport presets, file upload helpers, accessibility checks, retries, automatic waits.
    - Emphasis on test readability, modularity, and reliability.

CODE VA Testing Plan

Structured guidance for front-end/backend testing:

- Defines artifact management, exploratory/smoke/regression test types.
- Uses tools like GitHub, Docker, Jira, Confluence, and Cypress.

7. Recommended Patterns for Automated Testing

Aligning with VA's ATI and QAOps frameworks, teams should adopt:

- **Page Object Model (POM)** for UI abstraction.
- **BDD (Cucumber)** and **TestNG/JUnit** for structured test cases.
- **Factory or Fluent POM patterns** for scalable object creation.
- **Singleton pattern** for shared resources (e.g., WebDriver).
- **Facade pattern** for simplifying complex subsystem interactions (e.g., combined UI+API flows).
- **Accessibility scanning** via Java-A11y or Axe.

These match general automation best practices and are encouraged in ATI governance. Continuing to integrate new patterns (e.g., performance, service virtualization) is inherent in QAOps' evolution.

Summary of Engineering Guidelines

Area	Key Guidelines
Architecture	Adopt ATI 3-tier structure; standard test/build tools

Area	Key Guidelines
Frameworks	Java/Eclipse, Selenium, Appium, JUnit/TestNG, Cucumber
Coding & Test Practices	Enforce naming conventions, modularity, peer reviews, CI/CD integration
Governance	Use DOTS for orchestration; comply with TRM, FedRAMP, ATO
Tool Support	Utilize VATS for VistA, Cypress for frontend, JMeter for perf
QAOps Benefits	Shared codebase, tool standardization, accelerated onboarding, broad platform coverage

8. Automated Testing Adoption Checklist

Architecture and Framework Setup

- [ ] Use the ATI **3-tier architecture** (Automation Core → Page/Form Interfaces → Test Scripts).
- [ ] Use approved frameworks: **Java**, **JUnit/TestNG**, **Cucumber**, **Selenium**, **Appium/WinAppDriver**, **JMeter**, **Java-A11y**, **JAX-RS**.
- [ ] Confirm all tools are **TRM-approved** and **FedRAMP/ATO compliant** for SaaS.

Source Control and Coding Standards

- [ ] Follow standardized naming conventions for tests, objects, and data.
- [ ] Use **GitHub** with required pull-request reviews.
- [ ] Enforce branching strategy (feature branches → PR → mainline).
- [ ] Implement modular, reusable code (no inline selectors or repeated logic).

Test Design Patterns

- [ ] Implement **Page Object Model (POM)** for all UI automation.
- [ ] Use **BDD** where appropriate (Cucumber/Gherkin).
- [ ] Apply **Fluent POM**, **Factory**, or **Facade** patterns for complex workflows.
- [ ] Use **Singleton** for WebDriver or shared test resources.

Test Coverage Expectations

- [ ] Automate **smoke**, **regression**, **functional**, **API**, and **accessibility** tests.
- [ ] Integrate performance testing (JMeter) where load/response SLAs matter.
- [ ] Incorporate **Section 508** accessibility checks in every pipeline.

CI/CD Integration

- [ ] Integrate tests with VA's **MTS** tooling and CI/CD infrastructure (Jenkins/Azure/AWS).
- [ ] Configure automated triggers (pull requests, nightly builds, release gates).
- [ ] Centralize test results in approved tools (e.g., **Xray** in Jira).

Test Data and Environments

- [ ] Use controlled, reusable, and scrubbed test data sets.
- [ ] Avoid test interdependencies; designs must be self-contained and repeatable.
- [ ] Maintain environment-agnostic configurations (config files, environment variables).

Logging and Reporting

- [ ] Implement standardized logging format with timestamping.
- [ ] Capture screenshots or artifacts for failures.
- [ ] Publish results automatically to the test management system.

Onboarding and Knowledge Management

- [ ] Complete ATI onboarding steps: feasibility assessment → training → mentored development.
- [ ] Use shared QAOps libraries to avoid duplicating effort.
- [ ] Document test strategy, patterns, and framework usage in team Confluence/README.

Compliance and Governance

- [ ] Ensure all automation fits within ATI governance standards.
- [ ] Validate accessibility compliance in alignment with VA Section 508 requirements.
- [ ] Periodically review with QAOps for alignment and tool updates.

9. References

- GitHub Enterprise Technology Guideline: GitHub Technology Standard
- SharePoint Site for Demonstrations and ATI Manuals/Documentation: Quality Assurance Operations (QAOps) (sharepoint.com)
- MTS Service Portal: https://github.com/department-of-veterans-affairs/dots-tools-jenkins

Document Version Control

Date	Version	Change Details	By
1/18/2024	1.0	Initial version created with PDS input	OCTO/TAS
3/22/2025	2.0	Updated version for 2025 reflecting a link to ATI as a platform in CODE VA, and coordinated across PDS	OCTO/TAS
5/1/2026	3.0	Annual refresh with new formatting and minor updates, and new link to automated testing guidelines	OCTO/TAS

Disclaimer: This document serves both internal and external customers. Links displayed throughout this document may not be viewable to all users outside the VA domain. This document may also include links to websites outside VA control and jurisdiction. VA is not responsible for the privacy practices or the content of non-VA websites. We encourage you to review the privacy policy or terms and conditions of those sites to fully understand what information is collected and how it is used.

Statement of Endorsement: Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, and shall not be used for advertising or product endorsement purposes.