integration QA
MOVING WITH TECHNOLOGY

PARASOFT

Test and Migration Strategies for Oracle Middleware: A Case Study with Oracle Fusion

With Oracle middleware as the heartbeat of your application integration strategy, making a change to this infrastructure can have heart-stopping consequences. A migration to a new platform or a new version of the platform is even more precarious since new version features could cause undetected impacts. Leveraging simulation technology (service virtualization) and automated API (service) testing technologies in these scenarios can dramatically reduce the risk associated with change or migration.

There are three primary challenges associated with ensuring that Oracle middleware is achieving goals associated with system reliability, security, and performance:

- **Exercising a complete end-to-end workflow** is impeded by the high number and complexity of system dependencies involved in a typical end-to-end transaction.
- **Continuously testing that workflow** as the system is being developed or migrated is complicated by the challenge of constructing, initiating, and evaluating tests for such a specialized environment.
- **Realistic performance for dependent applications** is difficult to achieve in a test environment due to access constraints and system complexity, but it is nevertheless critical for achieving thorough and accurate validation—and thus more effective optimization— prior to deployment.

With comprehensive support for working within Oracle ecosystems, Parasoft's integrated API (service) testing and service virtualization (lab management) solution addresses these challenges, enabling you to effectively test your Oracle middleware as you're integrating and evolving it.

"Shift Left" End-to-End Testing

By providing a comprehensive, easy-to-use solution for enabling and continuously executing end-to-end testing through Oracle middleware, Parasoft enables organizations to test earlier, faster, and more completely.

Parasoft's industry-leading support message/protocol support (over 120 service and message types, including SOAP, JMS, MQ, Advanced Queuing and JDBC and others popular in Oracle environments) enables complex tests and validations to be rapidly constructed from an intuitive graphical test construction and management interface. Tests are constructed to support fully-automated, continuous regression testing—alerting you to unexpected changes while ignoring insignificant differences. Moreover, as systems evolve, automated intelligent updating helps you keep test assets in sync with changes.

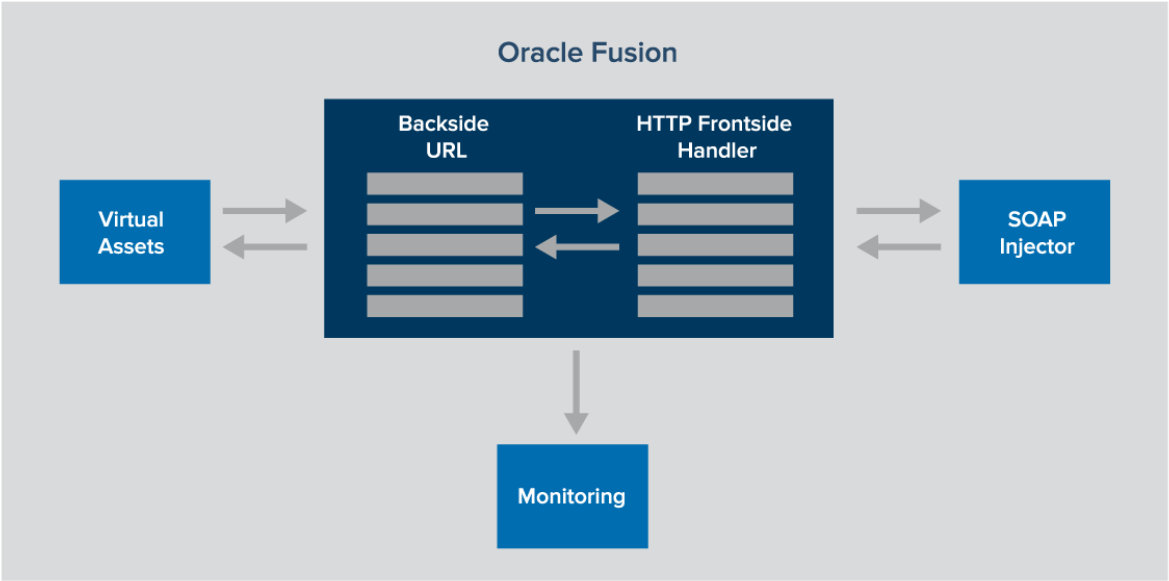
Moreover, if any dependencies (ERP, database, mainframes, third-party services, etc.) are not yet implemented or are not readily-available in a test environment, they can easily be replaced with "virtual assets" that assume the appropriate behavior, data, and performance profiles. For dependencies that are difficult to configure for specific testing needs (e.g., due to access constraints or the specialized nature of the technology), service virtualization gives developers and testers the

freedom to easily control their behavior as needed to complete a broad set of tests, including negative testing, corner cases, various performance scenarios, and so on. With a simulated test environment, team members and partners can have secure, 24/7 access to complete test environments.

When validating whether end-to-end transactions satisfy performance expectations, teams need to test and tune the AUT against realistic and consistent performance from dependencies. Yet, this can be particularly difficult to achieve in an Oracle ecosystem since your ability to configure dependencies may be limited by access constraints as well as system complexity. With service virtualization, the performance of each dependency is completely under your control. It's simple to configure and adjust the performance of "virtualized" dependencies to check various what-if scenarios—providing a fast and easy way to apply the various performance profiles needed to truly exercise the AUT.

Case Study: Testing Oracle Fusion Middleware SOAP Services

At a recent engagement for a telecommunications client, IntegrationQA implemented a Parasoft-driven framework to validate Oracle Fusion middleware SOAP requests/replies. The same library of pre-compiled audit checks, monitoring tools, test cases, and stress injectors applied here can be used as the foundation for running tests against *any* Oracle Fusion middleware service—at any organization.



Audit and Baseline

To establish a system for validating changes and baselining the middleware before executing each test cycle, IntegrationQA built an import facility that automatically extracts the Oracle Fusion configuration into Parasoft SOAtest. Checking the baseline provides a powerful tool to monitor change within the middleware and ensure that tests provide adequate coverage when changes occur. Using Parasoft SOAtest to audit the front-side handler definition (WSDL) validation of the schema, semantic and WS-I interoperability was checked automatically.

Isolating the Middleware

To reduce the dependency on any back-end service and provide detailed transactional information, IntegrationQA configured virtual assets to simulate the back-end requests/responses. "Ring fencing" the middleware with Parasoft virtual assets significantly lowered costs by reducing the number of test systems required. Additionally, reducing the dependency on downstream systems increased productivity by lowering downtime. Testing could begin much earlier with the new solution; in fact, it could begin well before the downstream system was complete.

API Testing, Regression Testing, and Continuous Testing

A set of tests covering each operation in the WSDL was generated in minutes using SOAtest's automated test generation capabilities. IntegrationQA then linked a pool of test data into these tests, enabling the organization to test a wide variety of scenarios from each API test.

For regression testing, Parasoft SOAtest's automated generation of regression controls was used to record the responses from the middleware. With the results between each test cycle automatically compared, the organization could rest assured that unexpected changes would be quickly exposed.

For continuous testing, the library of tests was configured to run automatically at regularly-scheduled intervals. This customer wanted to automate testing through the Windows Scheduler, but test execution can be initiated from *any* DevOps tool or Tests Management system that the organization prefers.

Performance Testing

The organization's API test suites were leveraged in Parasoft Load Test to perform load/performance testing against the middleware. No modification or additional configuration was required to "reuse" the API tests for performance testing.

Test Monitoring

IntegrationQA connected the Parasoft technology directly into Oracle Fusion's Message Reporting database so that transaction monitoring details could be readily reviewed. This gave the organization an easy means of troubleshooting any transaction through its lifecycle.

Determine Whether Migrations and Upgrades Are Implemented Correctly

Since flawless operation of Oracle middleware is so critical to so many business transactions, it's vital to ensure that migrations and updates do not unintentionally change or break any of the existing functionality that your system relies upon. Parasoft can help you assess the integrity of a migrated/upgraded middleware implementation by placing "bracers" around the legacy functionality,

then automatically comparing the new version against the baseline to assess whether the migration or upgrade has been implemented successfully.

Case Study: Migration from Oracle 10g to 11g

At recent engagements for clients in the Utilities and Telecom industries, IntegrationQA implemented a Parasoft-driven framework for testing Oracle middleware migration (from 10g to 11g). This framework, which can be used for any type of middleware, radically reduces the resources, time, and costs associated with middleware upgrades or migrations.

Both organizations' traditional migration efforts involved commissioning one or two end-to-end environments for several weeks to set up and configure the environment, then conduct manual API testing and end-to-end testing.

To help the customers ensure the integrity of the migration, IntegrationQA implemented a framework that leveraged Parasoft API testing, event monitoring, and service virtualization technologies to enable:

- Baselining of an existing environment and then verification of the new environment versus the baseline—in either an isolated or end-to-end environment.
- 100% coverage to ensure that any unexpected changes are exposed.

With the framework built and embedded into development's build and release process, the time required to verify business and technical changes was reduced to seconds. Moreover, since this framework could be controlled and "owned" by development, there was no need for additional testing resources to maintain, support, and verify the tests. This shift allowed the test teams to concentrate on 'real' issues that pose a high risk to the business.

About IntegrationQA

IntegrationQA is an independent software design, development-QA and test consultancy located in Wellington, New Zealand and Sydney, Australia. Established in 2009, after years of collective systems integration experience, led us to form our own practice to provide our accrued knowledge to clients throughout the Australia-New Zealand region. Our goal is to improve a client's IT infrastructure through better technical design, improved delivery practices and better software testing. Already, we have been able to provide significant cost reductions and improved efficiency to clients who in turn have dramatically altered their design, development and test approaches.

The IntegrationQA team boasts extensive experience working in Oracle environments, and has developed an extensive library of test and service virtualization artifacts that can be leveraged across customer environments. The team is comprised of highly-trained developers and architects who have put on a quality hat to approach Quality Assurance from a technical perspective. Their exceptional approach to testing is informed by their intimate knowledge of the systems being tested and migrated; the technical risks associated with migrations; and the appropriate test techniques to adopt. They specialize in building and implementing test frameworks supporting:

- Oracle Sun JCAPS (EAI, B2B)
- Oracle SOA Suite (OSB, BPEL, B2B)

- Oracle Rules Engine (OPA, Business Rules)
- Oracle WebLogic Integration
- Oracle WebLogic Server
- Oracle OAM
- Oracle Siebel
- IBM Datapower
- IBM Message Broker

Contacting IntegrationQA

NEW ZEALAND
Level 7
12-22 Johnston Street
Wellington 6011
T: +64 4 473 8535

AUSTRALIA
Level 39
2 Park Street
Sydney 2000
T: +61 2 803 53413

About Parasoft

Parasoft researches and develops software solutions that help organizations deliver defect-free software efficiently. By integrating Service Virtualization, Development Testing, and API testing, we reduce the time, effort, and cost of delivering secure, reliable, and compliant software. Parasoft's enterprise and embedded development solutions are the industry's most comprehensive—including static analysis, unit testing, requirements traceability, coverage analysis, functional & load testing, dev/test environment management, and more. The majority of Fortune 500 companies rely on Parasoft in order to produce top-quality software consistently and efficiently as they pursue agile, lean, DevOps, compliance, and safety-critical development initiatives.

Contacting Parasoft

USA	Phone: (888) 305-0041	Email: info@parasoft.com
NORDICS	Phone: +31-70-3922000	Email: info@parasoft.nl
GERMANY	Phone: +49 731 880309-0	Email: info-de@parasoft.com
POLAND	Phone: +48 12 290 91 01	Email: info-pl@parasoft.com
UK	Phone: +44 (0)208 263 6005	Email: sales@parasoft-uk.com
FRANCE	Phone: (33 1) 64 89 26 00	Email: sales@parasoft-fr.com
ITALY	Phone: (+39) 06 96 03 86 74	Email: c.soulat@parasoft-fr.com

OTHER See http://www.parasoft.com/contacts

Author Information

This paper was written by:

- Chris Wellington (chris.wellington@integrationqa.com), Managing Director at IntegrationQA
- Andrew Saunders (andrew.saunders@integrationqa.co.nz), Senior Consultant at IntegrationQA
- Wayne Ariola (wayne.ariola@parasoft.com), Chief Strategy Officer at Parasoft
- Cynthia Dunlop (cynthia.dunlop@parasoft.com), Lead Technical Writer at Parasoft