

An Oracle White Paper
March 2010

Application Quality Management

Introduction.....	1
Application Testing with Oracle Application Testing Suite	3
Oracle Test Manager	3
Oracle Functional Testing	5
Oracle Load Testing.....	6
Benefits of Application Testing Suite.....	7
Infrastructure Testing with Oracle Real Application Testing	8
SQL Performance Analyzer	9
Database Replay	10
Benefits of Real Application Testing.....	12
Secure Test Data Management with Oracle Data Masking Pack	13
Oracle Data Masking Pack.....	13
Benefits of Oracle Data Masking Pack.....	15
Conclusion.....	16

Introduction

Ensuring application quality and performance is one of the biggest challenges companies face as they deploy more applications and move more of their services online. It's also increasingly expensive as the pace of new application releases, upgrades and patches continues to grow and increases the need for testing. Oracle's Application Quality Management solutions provide high quality testing for all tiers of the application stack to help companies identify application quality and performance issues prior to deployment, reduce costs and ensure a positive experience for application end-users.

Companies have invested huge amounts in new applications to deliver better and more cost-effective services to their customers. But poor software quality can put these investments at risk. Studies have shown that more than 40% of software applications release with critical defects. And the cost to fix those defects in production is up to 100 times more expensive than in the development phase. In a customer satisfaction survey performed for Siebel customers, a correlation was shown between the amount of testing performed and customer satisfaction with the application. In every measure including overall product effectiveness, reliability, speed and scalability, customers that did more testing and more formalized testing had better results than those that did not test. This is why Application Quality Management is so important for maintaining business agility and ensuring service levels while also reducing costs and risk.

Oracle Enterprise Manager's Application Quality Management (AQM) solutions provide high quality testing for all tiers of the application stack. Thorough testing can help users identify application quality and performance issues prior to deployment. Testing is one of the most challenging and time consuming parts of successfully deploying an application, but it is also one of the most critical to the project's success. Oracle Enterprise Manager's AQM solutions provide a unique combination of test capabilities which enable users to:

- Test infrastructure changes: Real Application Testing is designed and optimized for testing database tier infrastructure changes using real application production workloads to validate database performance in your test environment.
- Test application changes: Application Testing Suite helps you ensure application quality and performance with complete end-to-end application testing solutions that allow you to automate functional & regression testing, execute load tests and manage the test process.
- Manage your test data and enable secure production-scale testing: Data Masking Pack helps you achieve security & compliance objectives by obfuscating sensitive data in your production databases so you can leverage them in your test environments.

Together these products provide a comprehensive solution for Application Quality Management.

Application Testing with Oracle Application Testing Suite

Rigorous application testing can help ensure a successful application deployment and a positive experience for your end users. This involves end-to-end testing of the entire application and supporting infrastructure to ensure that it meets requirements and performs as expected. Too often however, testing is left to the very end of the application development process with little or no planning done in advance to guarantee success. Application functionality is usually tested manually, on an ad-hoc basis and performance testing is often an afterthought. Testers and quality assurance teams are faced with both limited time and not enough resources to ensure critical applications will function properly. As application development inevitably slips, testing time is further reduced in an effort to meet oftentimes rigid release schedules.

Oracle Application Testing Suite is an integrated test solution that provides end-to-end testing capabilities for ensuring application quality, performance and reliability. Application Testing Suite (ATS) includes a suite of products for automated functional testing, load testing and test management of Web, packaged and Service Oriented Architecture-based applications. By helping you automate your test cases, test and tune application performance and better manage your test processes – ATS can help you deliver higher quality applications while also increasing the efficiency of your testing team.

There are three separately licensed products in the Oracle Application Testing Suite:

- Oracle Test Manager for documenting and managing the overall test process including test requirements, test cases and issues.
- Oracle Functional Testing for automating functional and regression testing of Web applications, packaged applications and Web Services.
- Oracle Load Testing for automated load testing of Web applications, packaged applications and Web Services.

ATS provides heterogeneous test capabilities for any Web application or Web Service, regardless of the server technology or platform it is built on. ATS also provides custom test “accelerators” for testing Oracle packaged applications like Oracle e-Business Suite and Siebel, providing more efficient and optimized testing. Using ATS, customers can help ensure the success of their mission-critical application deployments and upgrades.

Oracle Test Manager

Ensuring application quality in the face of tight release schedules and limited resources can be a major challenge. The only way to address this challenge is by having an effective framework, in place for planning, executing, and managing the testing process. Implementing an effective testing process up front will allow you to:

- Reduce Test Cycle Times
- Improve Testing Efficiency

- Promote Reusability of Test Cases
- Increase Visibility of the Testing Process

Oracle Test Manager provides a complete test process management solution – helping you manage all of your test cases, test requirements and issues from a central repository to improve the effectiveness of your test process. Users access Oracle Test Manager through a fully Web-based console that provides three integrated modules:

- The Requirements module allows testers to document and manage their test requirements before the testing process begins.
- The Tests module enables testers to create detailed test plans, document automated and manual test cases, execute these test cases and store results.
- The Issues module allows testers and developers to capture and share issues found during the testing process and then track their resolution.

Oracle Test Manager’s integrated reporting interface helps you provide end-to-end visibility into your test process. It’s also fully customizable to fit your test methodology. And because it is built on a scalable enterprise architecture based on Oracle Database and Weblogic Server, it provides unlimited scalability to meet your expanding needs.

The screenshot displays the Oracle Test Manager interface for a test case named "Verify single user login". The interface includes a navigation pane on the left, a main content area with test details, and a right-hand sidebar with attachments and requirements. The "Test Steps" table is as follows:

#	ACTION	EXPECTED RESULT	COMMENT	ATTACHMENTS
1	Go to http://demo.fmstocks.com/fmstocks/	Login page should appear with fields to specify "Email" and "Password"	May be pre-populated	
2	Enter "email" and "password" from test accounts list (see attached) and click "Login"	Should see main "Welcome" page and be able to access account information and research and trading areas		

The "Run History" table shows the following entry:

DATE	RAN BY	RESULT	SUMMARY
Thu, 2/07/02 12:31 PM	Default User	Failed	Failed step 2

Figure 1: Oracle Test Manager

Oracle Functional Testing

Functional testing helps you validate that your application's functionality meets end user business requirements. Whether you are deploying a brand new application or upgrading an existing application, functional testing is critical to ensure the application works as expected and to identify and address any issues prior to deployment. Functional testing requires both validating new functionality as well as testing existing functionality as part of an upgrade to make sure the application didn't "regress" in the latest release. This combination of functional and regression testing is key to ensuring overall application quality.

Manual testing accounts for the majority of functional & regression testing performed for all applications today. Manual testing typically involves a team of testers, developers, business analysts and possibly even end-users stepping through the application manually in a test or staging environment, to check that it works prior to being deployed to production. This type of testing leverages the hands-on knowledge and experiences of these testers. However, it can be very time consuming, expensive and inefficient - especially in the face of a major application rollout or a series of repeated application upgrade cycles that require continuous ongoing testing. Automated test tools can help you automate many of your test cases, extend test coverage to a greater percentage of your application and reduce the need for manual testing which saves you both time and money. Automated functional and regression testing can help you complement your manual testing activities and make your testers more efficient and effective.

Oracle Functional Testing is an automated functional & regression testing solution for Web, SOA and Oracle packaged applications. Oracle Functional Testing's OpenScript integrated scripting platform enables users to create automated test scripts that simulate complex business transactions. The same solution is used to create both automated functional test scripts and load test scripts, so users aren't forced to learn different tools and scripting languages for each task.

OpenScript provides an intuitive visual scripting interface that simplifies the scripting process combined with a powerful Java IDE that provides superior scripting extensibility for advanced users. With OpenScript, users can record automated test scripts by simply stepping through their Web transactions in a browser. OpenScript automatically captures all user actions and data inputs and accurately identifies browser objects. Users can then run their automated scripts to execute these transactions and leverage the graphical scripting interface to analyze playback results, parameterize script inputs and add custom test cases to validate application content. Load test scripts can also be created in OpenScript and then run in Oracle Load Testing across thousands of concurrent Virtual Users to validate application performance. Custom testing accelerators for Oracle E-Business Suite, Siebel and SOA Web Services provide customized automation capabilities for more efficient, optimized testing.

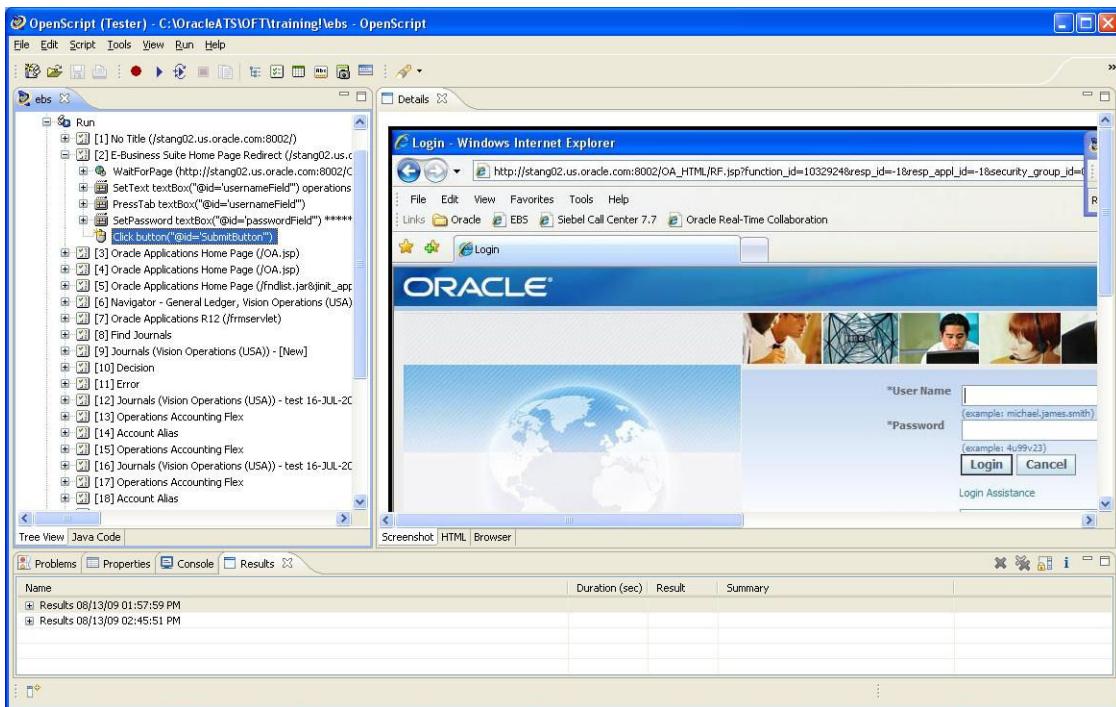


Figure 2: Oracle Functional Testing

Oracle Load Testing

Load testing can help you ensure that your application will perform and scale under real user workloads once it's deployed to production. This will help you assess that your application will be able to handle the number of concurrent users and resulting traffic during peak usage periods while maintaining acceptable performance and response times. It can also help you identify and address critical bottlenecks prior to deployment. Stress testing then allows you to test beyond the limits of normal operation and helps you assess the capacity and scalability of your application infrastructure.

In order to effectively analyze application performance and identify bottlenecks you need to be able to simulate production-level loads and accurately measure resulting application performance.

Performance tuning is also typically an iterative process that requires constant retesting as you address bottlenecks and make changes to your application and infrastructure. As a result, there really is no realistic manual alternative to run these tests without using an automated load testing tool.

Oracle Load Testing lets you run realistic load tests for Web, SOA and Oracle packaged applications – helping you simulate thousands of concurrent users and analyze the impact of production load levels on application performance. Users create load test scripts that automate key application workflows using Oracle's OpenScript integrated scripting platform. With OpenScript users can create load test scripts for any Web-based application regardless of the technology or platform it's built on. Integrated

load testing accelerator options then provide customized load testing capabilities for applications like Oracle e-Business Suite and Siebel as well as SOA-based Web Service interfaces.

Oracle Load Testing then provides an intuitive, fully-Web based console to configure these scripts to run across any number of concurrent users. Oracle Load Testing is deployed on the Oracle WebLogic Server and its intuitive Web interface allows you to configure your load test scenarios, run your tests and view graphs and reports to analyze application performance. Oracle Load Testing also includes integrated server monitors to help identify & resolve performance bottlenecks. This comprehensive set of infrastructure performance monitors can record in-depth performance metrics of Web servers, application servers, databases, and other infrastructure components during the load test. When this information is combined with the performance results gathered by the virtual users, developers have the real-time information needed to analyze and ensure optimal application performance during and after test execution.



Figure 3: Oracle Load Testing

Benefits of Application Testing Suite

Organizations that have used Oracle Application Testing Suite to automate functional & regression testing, execute load tests and manage their test process have realized significant benefits in the following areas:

- **Reduced Testing Time & Cost:** By automating their test cases and leveraging easy-to-use test solutions, organizations have been able to reduce the need for manual testing while making their test cycles more efficient and effective.
- **Increased Application Quality & Performance:** By leveraging test automation for both functional testing and load testing, organizations have been able to extend their test coverage, test their applications under production-level workloads and identify and address more functionality issues and performance bottlenecks prior to deployment, when they are less costly to fix.
- **Improved Control & Visibility over the Test Process:** By managing their testing from a centralized console, organizations have enabled their test teams to collaborate and share information while gaining more control and visibility over their test processes.

These benefits were realized by a major financial services company that was rolling out a new version of Siebel CRM. In addition to performing a major upgrade and adding new application functionality, the customer was also consolidating their server platform to fewer more powerful machines. A previous Siebel upgrade had been a challenge due to unexpected issues that had caused poor performance for end users, once the application was deployed to production. With Application Testing Suite, this customer was able to quickly automate key Siebel transactions in Oracle Functional Testing to create their load test scripts. They were then able to run these test scenarios across hundreds of concurrent users to simulate application usage in production. The tests identified application performance issues caused by excessive CPU usage in the database tier due to some costly, ad-hoc queries being executed. After addressing this performance bottleneck, they were able to re-run their load test to validate that the database was no longer an issue and application performance was acceptable. As a result of their thorough testing and tuning, the customer's production upgrade was a success and end-users reported positive feedback on the performance of the application.

Infrastructure Testing with Oracle Real Application Testing

Real Application Testing provides the highest quality load testing solution for the database stack complementing ATS's capabilities. Real Application Testing can be used for testing existing applications for system changes related to database stack or below. Some typical examples of supported system changes that routinely happen in a database operational environment are: Operating system and hardware upgrades, storage subsystem changes, database upgrade and/or patches, RAC instance addition, conversion to RAC, migration to Exadata V2, database parameter or optimizer related changes.

Real Application Testing includes two solutions to test the effect of system changes on real-world applications:

- **SQL Performance Analyzer (SPA)** to assess the impact of system changes on SQL response time by identifying any variation in SQL executions plans and performance statistics resulting from the change.

- Database Replay to effectively test system changes in test environments by replaying a full production workload on the test system to help determine the overall impact of change on the workload

Database Replay and SPA together provide a comprehensive, flexible, and end-to-end solution for assessing impact of database stack related changes. They enable businesses to fully assess the outcome of a system change in a test environment, take any corrective action if necessary, and then to introduce the change safely to production systems, minimizing the undesirable impact on them. Real Application Testing functionality is accessible both from Oracle Enterprise Manager and command-line APIs.

SQL Performance Analyzer

Changes in SQL execution plans due to routine system changes such as optimizer statistics refresh, schema changes, upgrades or patch set application, often severely impact production system performance and stability. Therefore, the ability to perform fine-grain SQL response time assessment with SPA and to fix any regressions is important to the smooth functioning of any application.

SPA runs the SQL statements in isolation and serial manner in before-change and after-change environments and provides a detailed change impact report showing SQL that have remained the same, improved and regressed. SPA functionality is integrated with database tuning solutions like SQL Tuning Advisor, and SQL Plan Management. As a result, SPA completely automates and simplifies the manual and time consuming process of identifying application SQL problems on even extremely large SQL workloads (hundreds of thousands of SQL statements) and also automating the remediation of any SQL regressions resulting from system changes.

The SPA report summarizes the change impact on entire workload as well as the net impact on individual SQL statements. Figure 1 below shows Oracle Enterprise Manager's SPA Task Result page for a completed SPA test run.

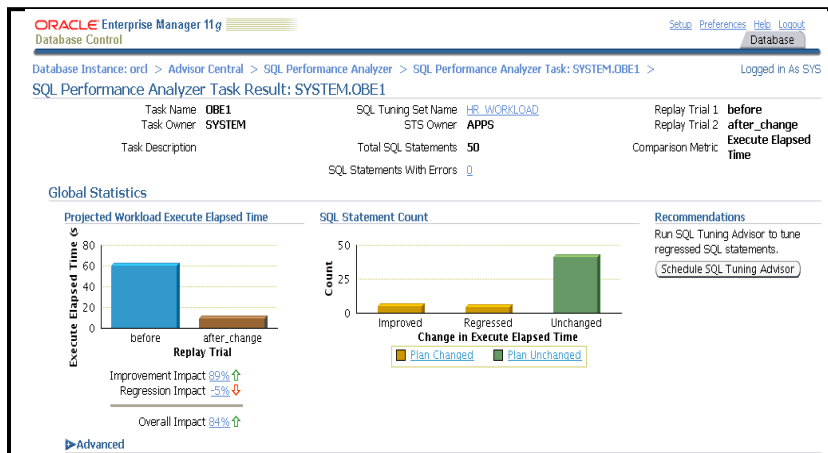


Figure 4: SQL Performance Analyzer Report

SPA testing covers all SELECT statements and query component of DML in the workload. As queries do not change the state of the data, SPA can even be used on production systems with appropriate resource and time limits during non-peak hours or maintenance windows. SPA testing can be scoped to private session attributes in a controlled manner to avoid impacting production users. For large terabyte-sized databases, some customers may be not able to provision a full-blown test system due to resource constraints. In such cases, SPA provides flexibility to test in production environment itself or also on a subset of the production database on test.

Besides providing testing capability in production environment, SPA testing also covers a broad range of database releases and use cases. For example, SPA can be for testing routine optimizer statistics refresh, Oracle Database 10.2.0.x to 10.2.0.y patch set application and Oracle Database 9i/10g to 10.2 and higher release upgrades. SPA testing can also be extended to home grown scripts, ATS or Database Replay. By capturing workloads into different STSs in two given environments (before and after change), and using the “Build from STS” trial method of SPA, one can understand the impact of system change on SQL workload.

Database Replay

Database Replay provides full workflow coverage and uses real production-scale workload that results in highest quality testing. Database Replay allows you to capture a production workload with negligible performance overhead and replay it on a test system with the exact timing, concurrency, and transaction characteristics of the original workload. By replaying real-production workload Database Replay provides complete assessment of the impact of the change including identifying undesired results - new contentions points or performance regressions. It also provides extensive analysis and reporting to help identify potential problems, such as new errors encountered and performance

divergence. Thus the task of assessing a system change using Database Replay is reduced from months to days.

Database Replay workload capture is performed at the database server level and therefore can be used to assess the impact of any system change below the database tier such as:

- Database upgrades, patches, parameter, schema changes, etc.
- Configuration changes such as conversion from a single instance to RAC, ASM
- Storage, network, interconnect changes
- Operating system, hardware migrations, patches, upgrades and parameter changes.

The Database Replay process can be broken down to 4 main steps:

- **Workload Capture** to record all requests made by external clients to Oracle Database, these include all relevant information about the client request, such as SQL text, bind values, and transaction information. Background activities and database scheduler jobs are not captured.
- **Workload Processing** to transform the captured data and create necessary metadata needed for replaying the workload.
- **Workload Replay** to submits calls to the database with the exact same timing and concurrency as in the capture system and puts the exact same load on the system as seen in the production environment. It is assumed that the test system is set up appropriately for replay purposes.
- **Analysis and Reporting** provides extensive information that helps understand impact of system change. Both high-level summary and detailed drill-down information in terms of errors, performance and data divergence are reported.

Figure 5 below shows Oracle Enterprise Manager's Database Replay Summary page of a completed workload replay.

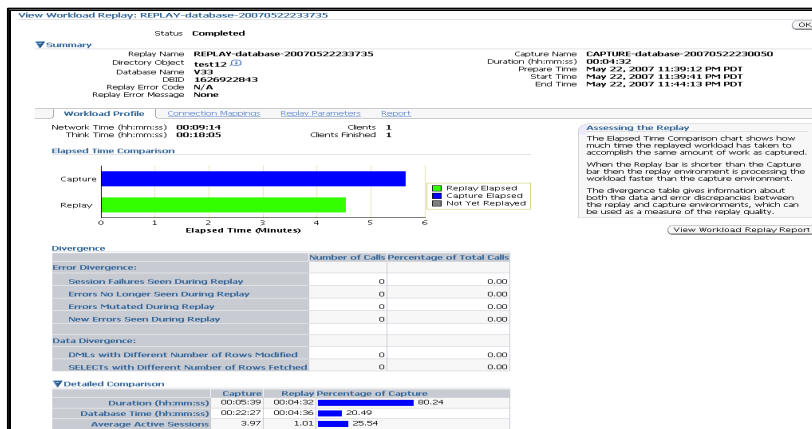


Figure 5: Database Replay Workload Replay Summary

Benefits of Real Application Testing

Businesses that have used Oracle Real Application Testing to test system changes have realized significant benefits in the following areas:

- **Highest quality testing:** By using real-production workload for testing, organizations have identified and corrected issues in test before production deployment. This had led to improved performance, SLAs, and stability of production systems.
- **Reduced IT costs:** By using Real Application Testing on an on-going basis, organizations have been able to reduce the burden on DBAs who previously had to be involved in fire fighting operational issues in production environment. Now these resources are directed towards more proactive and strategic part of the business.

A major retailer that used Real Application Testing to upgrade to Oracle Database 11g realized the above benefits. The customer's challenge in this case was to upgrade the mission critical database hosting the retail stores. The customer had previously used homegrown testing tools for upgrading from Oracle Database 8.1.7. They observed unpredictable performance following the 8.1.7 upgrade and numerous application changes were required to maintain system stability. With Real Application Testing use for Oracle Database 11g upgrade, the time to test was reduced by 50% from before and no application changes were necessary. And importantly, the production go-live was smooth and no surprises were noted in the last year of operations. The customer used SPA for testing SQL response time and finding the optimal database configuration setting for the workload. Database Replay was used for testing workload performance. The customer used SQL Profiles to improve application performance transparently. The retailer is now in a better position to accommodate future growth and plans to extend use of Real Application Testing for adopting new technologies like OLTP compression, TDE, etc. Thus, the retailer accomplished a smooth Oracle Database 11g upgrade with significantly reduced effort while eliminating the risk associated with such change.

Secure Test Data Management with Oracle Data Masking Pack

Enterprises have always shared data within and outside the organization for various business purposes. Database administrators (DBAs) in these enterprises copy production data into staging or test environments to allow in-house developers or offshore testers to perform application development and application testing. The problem with data sharing is that copies of production data often contain company confidential, sensitive or personally identifiable information, access to which is restricted by government regulations. Therefore, these enterprises run the risk of breaching sensitive information when sharing production data with application developers or software quality testers.

Oracle Data Masking Pack

Oracle Data Masking Pack helps reduce this risk by irreversibly replacing the original sensitive data with fictitious data so that production data can be shared safely with IT developers or offshore business partners. Oracle Data Masking Pack helps maintain the integrity of the application while masking data. Accessible via Oracle Enterprise Manager, this Management Pack provides end to end secure automation for provisioning test databases from production in compliance with regulations.

Sensitive Data Discovery and Application Integrity

Data may be sensitive for a variety of reasons, such as confidentiality (employee salary), regulatory (Sarbanes-Oxley or HIPAA compliance) or established business practices (PCI-DSS). Using Oracle Data Masking Pack's search capabilities, information security administrators can quickly search the database to identify sensitive data. In some applications, the same sensitive data is maintained in multiple tables related by referential (primary key-foreign key) relationships, e.g. employee numbers in a Human Resources application. Oracle Data Masking Pack discovers these relationships and masks all related data elements automatically while preserving referential relationships.

Comprehensive and Extensible Mask Library

Oracle Data Masking Pack provides a centralized library of out-of-the-box mask formats for common types of sensitive data, such as credit card numbers, phone numbers, national identifiers (social security number for US, national insurance number for UK). By leveraging the Format Library in Oracle Data Masking Pack, enterprises can apply data privacy rules to sensitive data across enterprise-wide databases from a single source and thus, ensure consistent compliance with regulations. Enterprises can also extend this library with their own mask formats to meet their specific data privacy and application requirements.

Sophisticated Masking Techniques

Oracle Data Masking Pack provides a variety of sophisticated masking techniques to meet application requirements while ensuring data privacy. These techniques ensure that applications continue to operate without errors after masking. For example,

- Condition-based masking: this technique makes it possible to apply different mask formats to the same data set depending on the rows that match the conditions. For example, applying different national identifier masks based on country of origin.
- Compound masking: this technique ensures that a set of related columns is masked as a group to ensure that the masked data across the related columns retain the same relationship, e.g. city, state, zip values need to be consistent after masking.
- Deterministic masking: this technique ensures repeatable masked values after a mask run. Enterprise may use this technique to ensure that certain values, e.g. a customer number gets masked to the same value across all databases.

Prior to mask execution, Oracle Data Masking Pack performs several pre-mask validation checks, such as validating that the mask formats matches the table data types, checking for space, to ensure that the masking process is error-free.

Unlike traditional masking processes that are typically slow, Oracle Data Masking Pack uses highly efficient parallelized bulk operations to replace the original sensitive data with masked data. Because the entire data masking process is done in place, enterprises can be assured of a greater sense of security knowing that the sensitive data would never leave the database during the masking process.

Oracle Data Masking Pack is also integrated with Oracle Provisioning and Patch Automation Pack in Oracle Enterprise Manager to clone-and-mask via a single workflow. The secure high performance nature of Oracle Data Masking combined with the end-to-end workflow ensures that enterprise can provision test systems from production rapidly instead of days or weeks that it would with separate manual processes.

Optimized for Oracle Databases

Oracle Data Masking Pack leverages key capabilities in Oracle databases to enhance the overall manageability of the masking solution. Some of these include:

- Flashback: Administrators can optionally configure Oracle databases to enable flashback to a pre-masked state if they encounter problems with the masked data.
- PL/SQL: Unlike other solutions, Oracle Data Masking Pack generates DBA-friendly PL/SQL that allows DBAs to tailor the masking process to their needs. This PL/SQL script can also be easily integrated into any cloning process.

Select	Format	Data Type	Sample	Description	Owner
<input checked="" type="radio"/>	American Express Credit Card Number	Character	3415423650700120	~10 billion unique American Express credit card numbers	SYSMAN
<input type="radio"/>	Discover Card Credit Card Number	Character	6011379985694035	~10 billion unique Discover Card credit card numbers	SYSMAN
<input type="radio"/>	MasterCard Credit Card Number	Character	5452990103307652	~10 billion unique MasterCard credit card numbers	SYSMAN
<input type="radio"/>	Visa Credit Card Number	Character	4532133441705009	~10 billion unique Visa credit card numbers	SYSMAN
<input type="radio"/>	Generic Credit Card Number	Character	6011681084243014	~10 billion unique generic credit card numbers	SYSMAN
<input type="radio"/>	Generic Credit Card Number Formatted	Character	4485-7405-3460-6006	~10 billion unique generic credit card numbers	SYSMAN
<input type="radio"/>	National Insurance Number Formatted	Character	GG 44 77 64 A	Generates unique UK National Insurance Numbers	SYSMAN
<input type="radio"/>	Social Insurance Number	Character	482919701	~1 billion unique Canadian Social Insurance Numbers	SYSMAN
<input type="radio"/>	Social Insurance Number Formatted	Character	601-572-803	~1 billion unique Canadian Social Insurance Numbers	SYSMAN
<input type="radio"/>	Social Security Number	Character	419428920	~718 million unique US Social Security Numbers	SYSMAN
<input type="radio"/>	Social Security Number Formatted	Character	143-87-2130	~718 million unique US Social Security Numbers	SYSMAN
<input type="radio"/>	ISBN (Ten Digit)	Character	4405031509	~1 billion unique ISBN numbers	SYSMAN
<input type="radio"/>	ISBN (Ten Digit) Formatted	Character	8-71-618980-9	~1 billion unique ISBN numbers	SYSMAN
<input type="radio"/>	ISBN (Thirteen Digit)	Character	9793811520836	~2 billion unique ISBN numbers	SYSMAN
<input type="radio"/>	ISBN (Thirteen Digit) Formatted	Character	979-0-744630-63-2	~2 billion unique ISBN numbers	SYSMAN
<input type="radio"/>	UPC Number	Character	355939020101	~100 billion UPC numbers	SYSMAN
<input type="radio"/>	UPC Number Formatted	Character	5-71361-97790-9	~100 billion UPC numbers	SYSMAN
<input type="radio"/>	USA Phone Number	Character	4352194430	~2.7 billion unique USA phone numbers	SYSMAN
<input type="radio"/>	USA Phone Number Formatted	Character	631-986-2400	~2.7 billion unique USA phone numbers	SYSMAN

View | Create Like | Edit | Delete

Number Range: 1 - 279 (both inclusive); Digits Length Range: 7 - 7; Function Name: DBSNMP.DM_FMTLIB.MGMT_DM_GEN_PH_USA_FH

Home | **Targets** | Reports | Setup | Preferences | Help | Logout

Figure 6: Data Masking Pack

Benefits of Oracle Data Masking Pack

Organizations that have implemented Oracle Data Masking Pack to protect sensitive data in test and development environment have realized significant benefits in the following areas:

- **Compliance:** By protecting sensitive information when sharing production data with developers and testers, organizations have able to ensure that non-production databases have remained compliant with IT security policies while enabling developers to conduct production-class testing.
- **Automation:** By automating the masking process, organizations have been able to reduce the burden on DBAs who previously had to maintain manually-developed masking scripts.

These benefits were realized by a major global telecommunications products company that implemented Oracle Data Masking Pack. Their database administrators (DBAs) had developed custom scripts to mask sensitive data in the test and development environments of their human resources (HR) application. As the company was growing and offering new services, their IT infrastructure was also growing thus placing an increased burden on their DBAs. By implementing Oracle Data Masking Pack, the organization was able to use the role-based separation of duties to allow the HR analysts to define the security policies for masking sensitive data. The DBAs then automated the implementation of these masking policies when provisioning new test or development environments. Thus, the

telecommunications company was able to allow business users to ensure compliance of their non-production environments while eliminating another manual task for the DBAs through automation.

Conclusion

Ensuring the quality and performance of your enterprise applications requires a comprehensive approach to application quality management. This requires thorough testing of all tiers of the application stack prior to deployment. This includes testing both applications and infrastructure, for new application deployments as well as upgrades of existing applications. Comprehensive testing requires validating both application functionality as well as performance under real-world operating conditions. And to maximize efficiency, it's important to have an effective framework in place to plan and manage your test processes and to leverage test automation to reduce the need for manual testing.

Oracle Enterprise Manager provides a comprehensive set of Application Quality Management solutions which include Application Testing Suite, Real Application Testing and Data Masking Pack. This best-of-breed Oracle AQM offering allows you to test the entire stack from application to database. It provides the only load testing solution on the market that combines both real and synthetic workload testing. And it enables high quality and secure testing with the lowest risk of change. Oracle AQM solutions provide a heterogeneous testing solution that is also optimized for testing Oracle applications. Oracle AQM will help you reduce test cycle times and costs while increasing the quality and performance of your applications.



Application Quality Management
March 2010

Author: Joe Fernandes

Contributing Authors: Jagan Athreya, Prabhaker
Gongloor

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2010, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0110