An Oracle White Paper
June 2013

Data Masking Best Practice

Executive Overview

Many organizations inadvertently breach information when they routinely copy sensitive or regulated production data into non-production environments. As a result data in non-production environment has increasingly become the target of cyber criminals and can be lost or stolen. Just like data breaches in production environments, data breaches in non-production environments can cause millions of dollars to remediate and cause irreparable harm to reputation and brand.

With Oracle Data Masking, sensitive and valuable information can be replaced with realistic values. This allows data to be safely used in non-production and incompliance with regulatory requirements such as Sarbanes-Oxley, PCI DSS, HIPAA and as well as numerous other laws and regulations.

This paper describes the best practices for deploying Oracle Data Masking to protect sensitive information in Oracle databases and other heterogeneous databases such as IBM DB2, Microsoft SQLServer.

Introduction – Why mask data?

Enterprises share data from their production applications with other users for a variety of business needs.

- Most organizations if not all copy production data into test and development environments to allow system administrators to test upgrades, patches and fixes.

- Businesses to stay competitive require new and improved functionality in existing production applications. As a result application developers require an environment mimicking close to that of production to build and test the new functionality ensuring that the existing functionality does not break.

- Retail companies share customer point-of-sale data with market researchers to analyze customer buying patterns

- Pharmaceutical or healthcare organizations share patient data with medical researchers to assess the efficiency of clinical trials or medical treatments

.As a result of the above, organizations copy tens of millions of sensitive customer and consumer data to non-production environments and very few companies do anything to protect this data, even when sharing with outsourcers and third parties.

Numerous industry studies on data privacy have concluded that companies do not prevent this sensitive data from coming in the hands of wrong-doers. Almost 1 out of 4 companies responded that this live data had been lost or stolen and 50% said that they had no way of knowing if the data in non-production environment had been compromised.

To protect and enforce against risk of compromising critical and confidential information, laws, regulations and business policies have been instigated by government and enterprise. As an example, The Health Insurance Portability and Accountability Act require the protection and confidential handling of protected health information. Also, the Payment Card Industry (PCI) Data Security Standard (DSS) which is enforced by Visa and Master Card was development to encourage and enhance cardholder data security and to facilitate the broad adoption of consistent data security measures globally. PCI DSS provides a baseline of technical and operational requirements designed to protect cardholder data.

With the explosion of E-commerce , business do not typically encrypt data beyond what is required because doing so degrades the performance of the production environment and also hamper non-production activities. Two thousand and eleven proved to be an unprecedented year for headlines about major database break-ins at Sony, Google, Bank of America, RSA, Lockheed, Epsilon, Nasdaq Directors Desk and the US Chamber of Commerce among many others. Hence, just like data breaches in production environments that has been reported in 2011, data breaches in non-production environment can cause irreparable harm to the reputation and brand. Enterprises that has spent over a decade in building their reputation, can painfully take so many steps backwards due to a single incident. Security experts and technologists point to several developments that suggest the pattern is likely to continue in 2013 as it did with Zappos in 2012.

Protecting vital company information in non-production environment has become one of the foremost critical tasks over the recent years. With Oracle Data Masking pack sensitive and valuable information can be replaced with realistic values. This allows production data to be safely used for development, testing, outsource partners and off-shore partners or other non production purposes.

The Challenges of masking non-production environments

Organizations have taken these threats seriously and have set out to address these issues as quickly as possible knowing the ramifications. However, the idea of simply removing sensitive information from non-production environment seems to be simple, it can pose serious challenges in various aspects.

Some of the immediate challenges are identifying sensitive information. What defines sensitive information? Where does it reside? How is it referenced? Applications have become very complex and integrated. Knowing where the sensitive information resides and what applications are referencing this information becomes a daunting task. Culminated with the ever evolving application, the challenge also becomes maintaining meta-data knowledge of the application architecture though-out its lifecycle.

Once sensitive information has been identified, the process of masking while maintaining application integrity becomes paramount. Simply changing the value will inadvertently break the application that is being used to test, develop or upgrade. As an example masking a part of a customer's address, such as zip without consideration of city and state, may render the application unusable. Hence developing or testing becomes if not impossible, unreliable.

Auditing is another challenge that is considered seriously. Knowing who changed what and when becomes an important business control requirement to prove compliance with regulations and laws. To implement these types of controls, the challenge becomes separations of duties, role based permissions and the ability to report on these activities.

Databases are becoming very large and the frequency of requests for a secure non-production environment to be available has drastically increased over the years. The reason for this increase is for business to develop newer and better applications which services their customers at a faster pace to stay competitive. A masking process needs to have acceptable performance and reliability.

And finally having a flexible solution that can evolve with the application and extend to other applications within an enterprise becomes an important challenge to address.

As a result of these challenges, unfortunately organizations have tried to address these issues with custom hand-crafted solutions or repurposed existing data manipulation tools within the enterprise to solve this problem of sharing sensitive information with non-production users. Take for example, the most common solution: database scripts. At first glance, an advantage of the database scripts approach would appear that they specifically address the unique privacy needs of a particular database that they were designed for. They may have even been tuned by the DBA to run at their fastest

Let’s look at the issues with this approach.

1. **Reusability**: Because of the tight association between a script and the associated database, these scripts would have to be re-written from scratch if applied to another database. There are no common capabilities in a script that can be easily leveraged across other databases.

2. **Transparency**: Since scripts tend to be monolithic programs, auditors have no transparency into the masking procedures used in the scripts. The auditors would find it extremely difficult to offer any recommendation on whether the masking process built into a script is secure and offers the enterprise the appropriate degree of protection.

3. **Maintainability**: When these enterprise applications are upgraded, new tables and columns containing sensitive data may be added as a part of the upgrade process. With a script-based approach, the entire script has to be revisited and updated to accommodate new tables and columns added as a part of an application patch or an upgrade.

Implementing Data Masking

With these enterprise challenges in mind, Oracle has development a comprehensive 4-step approach to implementing data masking via Oracle Data Masking Pack called: Find, Assess, Secure and Test (F.A.S.T). These steps are:

- **Find**: This phase involves identifying and cataloging sensitive or regulated data across the entire enterprise. Typically carried out by business or security analysts, the goal of this exercise is to come up with the comprehensive list of sensitive data elements specific to the organization and discover the associated tables, columns and relationships across enterprise databases that contain the sensitive data.

- **Assess**: In this phase, developers or DBAs in conjunction with business or security analysts identify the masking algorithms that represent the optimal techniques to replace the original sensitive data. Developers can leverage the existing masking library or extend it with their own masking routines.

- **Secure**: This and the next step may be iterative. The security administrator executes the masking process to secure the sensitive data during masking trials. Once the masking process has completed and has been verified, the DBA then hands over the environment to the application testers.

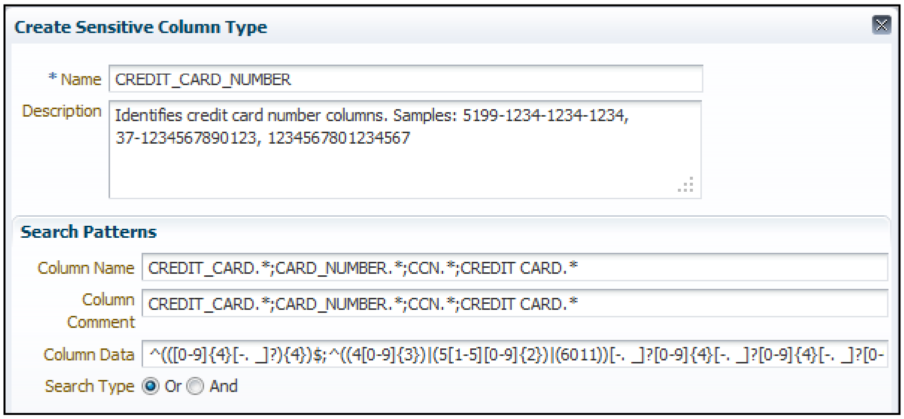
- **Test**: In the final step, the production users execute application processes to test whether the resulting masked data can be turned over to the other non-production users. If the masking routines need to be tweaked further, the DBA restores the database to the pre-masked state, fixes the masking algorithms and re-executes the masking process.

We will now dive deep into the individual steps and cover the best practice for enterprises to secure their non-production environment effectively using Oracle Data Masking.

Find: Comprehensive Enterprise-wide Discovery of Sensitive Data

To begin the process of masking data, the data elements that need to be masked in the application must be identified. The first step that any organization must take is to determine what is sensitive. This is because sensitive data is related to specific government regulations and industry standards that cover how the data can be used or shared. Thus, the first step is for the security administrator to publish what constitute sensitive data and get agreement from the company’s compliance or risk officers.

Once the sensitive elements have been decided upon, the next step involved is locating, or finding these sensitive elements in the databases. With Oracle Data Masking Pack, the Data Discovery and Modeling capability in Oracle Enterprise Manager, enterprises can define data pattern search criteria’s allowing security administrators to locate these sensitive elements. For example data pattern's such has 15- or 16-digits for credit card numbers or 9-digit formatted US social security numbers.



Create Sensitive Column Type

* Name: CREDIT_CARD_NUMBER

Description: Identifies credit card number columns. Samples: 5199-1234-1234-1234, 37-1234567890123, 1234567801234567

Search Patterns

Column Name: CREDIT_CARD.*;CARD_NUMBER.*;CCN.*;CREDIT CARD.*

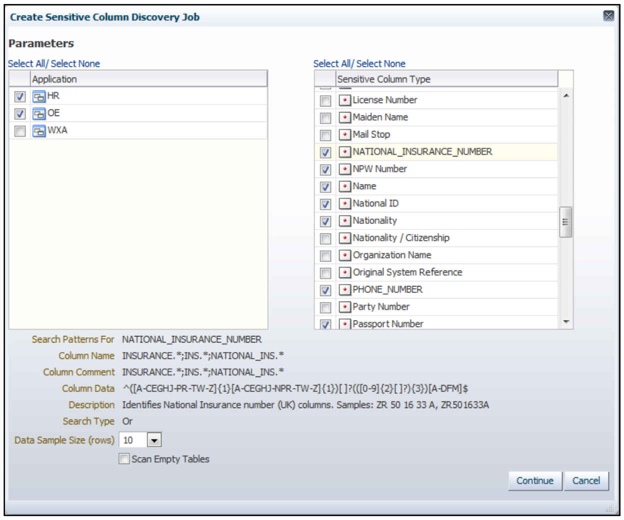
Column Comment: CREDIT_CARD.*;CARD_NUMBER.*;CCN.*;CREDIT CARD.*

Column Data: ^(([0-9]{4}[-. _]?){4})$;^((4[0-9]{3})|(5[1-5][0-9]{2})|(6011))[-. _]?[0-9]{4}[-. _]?[0-9]{4}[-. _]?[0-

Search Type: ● Or ○ And

Figure 1. Sensitive Column Type definition

Once all the sensitive elements have been defined, the system administrator will then schedule a discovery job in Enterprise Manager which will introspect the database application of concern.



Create Sensitive Column Discovery Job

Parameters

Select All/ Select None

	Application
☑	HR
☑	OE
□	WXA

Select All/ Select None

	Sensitive Column Type
□	License Number
□	Maiden Name
□	Mail Stop
☑	NATIONAL_INSURANCE_NUMBER
☑	NPW Number
☑	Name
☑	National ID
☑	Nationality
□	Nationality / Citizenship
□	Organization Name
□	Original System Reference
☑	PHONE_NUMBER
□	Party Number
☑	Passport Number

Search Patterns For: NATIONAL_INSURANCE_NUMBER

Column Name: INSURANCE.*;INS.*;NATIONAL_INS.*

Column Comment: INSURANCE.*;INS.*;NATIONAL_INS.*

Column Data: ^([A-CEGHJ-PR-TW-Z]{1}[A-CEGHJ-NPR-TW-Z]{1})[ ]?(([0-9]{2}[ ]?){3})[A-DFM]$

Description: Identifies National Insurance number (UK) columns. Samples: ZR 50 16 33 A, ZR501633A

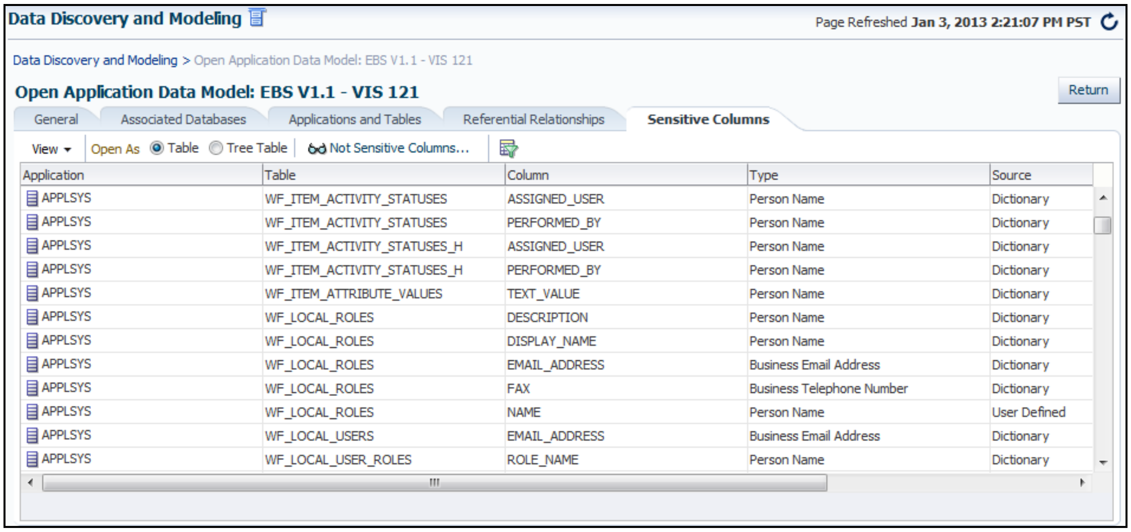
Search Type: Or

Data Sample Size (rows): 10

□ Scan Empty Tables

Figure 2. Sensitive column discovery

The search results returned are then ranked based on the probability of a match allowing the security administrator to designate the column as sensitive for inclusion in the masking process or not sensitive for exclusion from future ad hoc pattern searches.


Data Discovery and Modeling
Page Refreshed Jan 3, 2013 2:21:07 PM PST
Data Discovery and Modeling > Open Application Data Model: EBS V1.1 - VIS 121
Open Application Data Model: EBS V1.1 - VIS 121
Return
General
Associated Databases
Applications and Tables
Referential Relationships
Sensitive Columns
View
Open As
Table
Tree Table
Not Sensitive Columns...

Application	Table	Column	Type	Source
APPLSYS	WF_ITEM_ACTIVITY_STATUSES	ASSIGNED_USER	Person Name	Dictionary
APPLSYS	WF_ITEM_ACTIVITY_STATUSES	PERFORMED_BY	Person Name	Dictionary
APPLSYS	WF_ITEM_ACTIVITY_STATUSES_H	ASSIGNED_USER	Person Name	Dictionary
APPLSYS	WF_ITEM_ACTIVITY_STATUSES_H	PERFORMED_BY	Person Name	Dictionary
APPLSYS	WF_ITEM_ATTRIBUTE_VALUES	TEXT_VALUE	Person Name	Dictionary
APPLSYS	WF_LOCAL_ROLES	DESCRIPTION	Person Name	Dictionary
APPLSYS	WF_LOCAL_ROLES	DISPLAY_NAME	Person Name	Dictionary
APPLSYS	WF_LOCAL_ROLES	EMAIL_ADDRESS	Business Email Address	Dictionary
APPLSYS	WF_LOCAL_ROLES	FAX	Business Telephone Number	Dictionary
APPLSYS	WF_LOCAL_ROLES	NAME	Person Name	User Defined
APPLSYS	WF_LOCAL_USERS	EMAIL_ADDRESS	Business Email Address	Dictionary
APPLSYS	WF_LOCAL_USER_ROLES	ROLE_NAME	Person Name	Dictionary

Figure 3. E-Business Suite template pre-defined sensitive columns

Defining and identifying sensitive data to mask is only part of the solution. It is also, important to ensure data integrity to maintain correct application behavior after masking and to ensure integrity you must consider referential data relationships.

Today's relational databases store data in tables related by certain key columns called primary key columns to allow for efficient storage of application data without having to duplicate data. For example, an EMPLOYEE_ID generated from a human capital management (HCM) application may be used in sales force automation (SFA) application tables using foreign key columns, in a database or across databases.

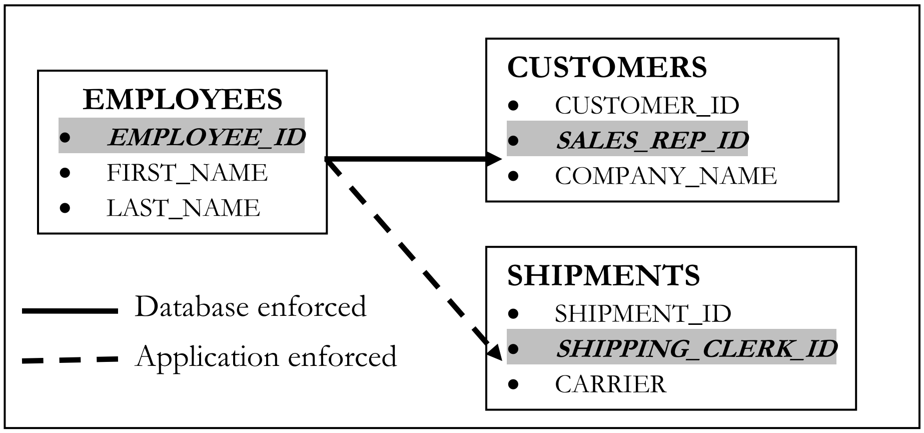


Figure 4. The Importance of Referential Integrity

Oracle Data Masking Pack automatically detects data dependencies such as foreign key constraints ensuring referential integrity. What this means, is that as part of the discovery of sensitive columns, Data Discovery and Modeling will also introspect database enforced relationships and stores them with the sensitive columns. This logical containment of entities, their relationships and the sensitive columns for an application or many applications is referred to as the Application Data Model (ADM) and is stored in the Enterprise Manager repository.

Application Data Model provides a robust mechanism by which security administrators can maintain this application meta-data knowledge and repeatedly use it as the building blocks for masking sensitive data in an enterprise, as we will mention later in this document. Application Data Model also allows the system administrator to maintain the sensitive columns very easily throughout the lifecycle of the application. Additionally, any modifications to the data model is simply performed my editing the data model within Enterprise Manager.

In addition to the above easy-to-use mechanism for isolating sensitive data elements and understanding the relationships, Oracle Data Masking Pack delivers meta-data knowledge of packaged applications in the form of templates that allow enterprises to quickly get started in masking sensitive data. Let us take Oracle E-Business Suite Data Masking Template as an example. This template contains out-of-the-box, meta-data knowledge of the E-Business Suite architecture and sensitive columns. It covers all product families shipped with Oracle E-Business Suite and contains PII and other sensitive personal meta-data knowledge associated with users. Additionally, EBusiness Suite, like any other packaged application, requires certain system users not be masked. This allows for the continued use of the application after masking so that developers, QA or non-production users can use the environment. This capability is provided for in the template and allows customers to add exempt users from masking by inserting into the table FND_USER_MASKING_EXEMPTIONS table. Follow the steps outlined in MOS Note: 1481916.1

Some of the attributes that are masked in the Oracle E-Business Suite Data Masking template are (For a complete list please refer to MOS Note 1481916.1):

TABLE 1. PARTIAL LIST OF ATTRIBUTES MASKED IN ORACLE E-BUSINESS SUITE DATA MASKING TEMPLATE

SENSITIVE ATTRIBUTE	PII ATTRIBUTE
Compensation	Person Name

Employment Details	Employee Number
Nationality / Citizenship	Account Name
Health Information	GPS Location
Session Information	National Identifier
Audit Information	Deriver License Number

Note that in the E-Business Suite Template, financial data such as results, forecasts, design specifications, unstructured data such as Descriptive Flex fields, notes, attachments and internal primary keys such as user_id or person_id are not masked

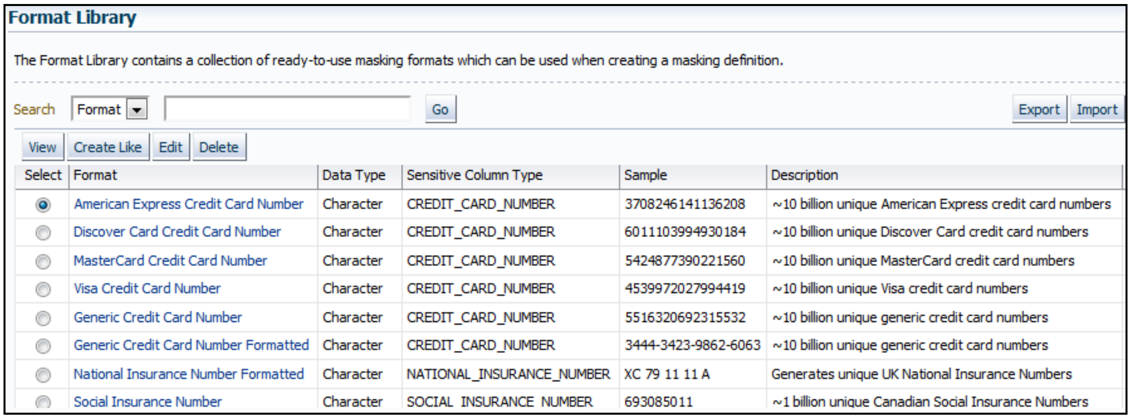
The templates that are currently available at time of writing are Oracle E-Business Suite and Oracle Fusions Applications. PeopleSoft data masking templates are scheduled to be in the next PSFT release.

Assess: Extensive out-of-the-box optimal masking algorithms

Data masking is in general a trade-off between security and reproducibility. A test database that is identical to the production database is 100% in terms of reproducibility and 0% in terms of security as it exposes the original data to non-production users. Masking technique where data in sensitive columns is replaced with a single fixed value is 100% in terms of security and 0% in terms of reproducibility. When considering various masking techniques, it is important to consider this trade-off in mind when selecting the masking algorithms.

After the Application Data Model has been built and the security administrator has identified all the sensitive columns that are required to be masked per rules and regulations, developers and or DBAs in conjunction with business security analysts identify the masking algorithms that represent the optimal techniques to replace the original sensitive data.

Oracle Data Masking 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, enterprise can apply data privacy rules to sensitive data across enterprise-wide databases from a single source and thus, ensure consistent compliance with regulations. Enterprise can also extend this library with their own mask formats to meet their specific data privacy and application requirements.



Format Library

The Format Library contains a collection of ready-to-use masking formats which can be used when creating a masking definition.

Select	Format	Data Type	Sensitive Column Type	Sample	Description
●	American Express Credit Card Number	Character	CREDIT_CARD_NUMBER	3708246141136208	~10 billion unique American Express credit card numbers
○	Discover Card Credit Card Number	Character	CREDIT_CARD_NUMBER	6011103994930184	~10 billion unique Discover Card credit card numbers
○	MasterCard Credit Card Number	Character	CREDIT_CARD_NUMBER	5424877390221560	~10 billion unique MasterCard credit card numbers
○	Visa Credit Card Number	Character	CREDIT_CARD_NUMBER	4539972027994419	~10 billion unique Visa credit card numbers
○	Generic Credit Card Number	Character	CREDIT_CARD_NUMBER	5516320692315532	~10 billion unique generic credit card numbers
○	Generic Credit Card Number Formatted	Character	CREDIT_CARD_NUMBER	3444-3423-9862-6063	~10 billion unique generic credit card numbers
○	National Insurance Number Formatted	Character	NATIONAL_INSURANCE_NUMBER	XC 79 11 11 A	Generates unique UK National Insurance Numbers
○	Social Insurance Number	Character	SOCIAL_INSURANCE_NUMBER	693085011	~1 billion unique Canadian Social Insurance Numbers

Figure 5. Central Format Library

Oracle Data Masking also provides mask primitives, which serve as building blocks to allow the creation of nearly unlimited custom mask formats ranging from numeric, alphabetic or date/time based. Recognizing that the real-world masking needs require a high degree of flexibility, Oracle Data Masking allows security administrators to create user-defined-masks. These user-defined masks, written in PL/SQL, let administrators create unique mask formats for sensitive data, e.g. generating a unique email address from fictitious first and last names to allow business applications to send test notifications to fictitious email addresses.

Commonly, enterprises require advanced masking rules to be used to maintain privacy of sensitive data and allow the application to continue functioning in a realistic manner. Oracle Data Masking provides a variety of sophisticated masking techniques to meet these application requirements while ensuring data privacy. Let us have a look at some

Sophisticated Masking Techniques

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. As an example, it is common that a global business will have employees in different countries, such as the US and UK. In the US, the national identifier of a person is the social security number which is 9 digits long where as in UK it is the national insurance number which is 9 alphanumeric long. Both of these may reside in the same column, and after masking the applications may need to keep the same characteristics of the data yet masked to ensure correct functionality. To determine what the format the data should be masked in, condition based masking of Oracle Data Masking Pack allows the functionality to check a country code and use the appropriate algorithm on the national identifier column

- **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. We will elaborate on this technique as it is a very common use case.

- **Key-based reversible masking**: when businesses need to send their data to a 3rd party for analysis, reporting or any other business process, this technique transforms the original data into a masked representation of itself using a secure key-based reversible masking function. Once the data is recovered from the 3rd party, the business can recover the original data by reversing the masking using the same key

Deterministic Masking

Deterministic masking is an important masking technique that enterprises must consider when masking key data that is referenced across multiple applications. Take, for example, three applications: a human capital management application, a customer relationship management application and a sales data warehouse. There are some key fields such as EMPLOYEE ID referenced in all three applications and needs to be masked in the corresponding test systems: a employee identifier for each employee in the human resources management application, customer service representative identifiers, which may also be EMPLOYEE IDs, in the customer relationship management application and sales representative IDs, which may be EMPLOYEE IDs in the sales data warehouse.

To ensure that data relationships are preserved across systems even as privacy-related elements are removed, deterministic masking techniques ensure that data gets masked consistently across the various systems. It is vital that deterministic masking techniques used produce the replacement masked value consistently and yet in a manner that the original data cannot be derived from the masked value.

One way to think of these deterministic masking techniques is as a function that is applied on the original value to generate a unique value consistently that has the same format, type and characteristics as the original value, e.g. a deterministic function f(x) where f(x1) will always produce y1 for a given value x1. In order for the deterministic masking to be applied successfully, it is important that the function f(x) not be reversible, i.e. the inverse function f-1(y1) should not produce x1 to ensure the security of the original sensitive data.

Deterministic masking techniques can be used with mathematical entries, e.g. social security numbers or credit card numbers, as well as with text entries, e.g., to generate names. For example, organizations may require that names always get masked to the same set of masked names to ensure consistency of data across runs. Testers may find it disruptive if the underlying data used for testing is changed by production refreshes and they could no longer locate certain types of employees or customer records that were examples for specific test cases. Thus, enterprises can use the deterministic masking functions provided by Oracle Data Masking to consistently generate the same replacement mask value for any type of sensitive data element.

Deterministic masking becomes extremely critical when testing data feeds coming from external systems, such as employee expense data provided by credit card companies. In production environments, the feed containing real credit card numbers are processed by the accounts payable application containing employee’s matching credit card information and are used to reconcile employee expenses. In test systems, the employee credit card numbers have been obfuscated and can no longer be matched against the data in the flat files containing the employee’s real credit card number. To address this requirement, enterprises pre-load the flat file containing data using tools such as SQL*Loader, into standard tables, then mask the sensitive columns using deterministic masking provided by Oracle Data Masking and then extract the masked data back into flat file. Now, the application will be able to process the flat files correctly just as they would have been in Production systems.

Packaged Application Data Masking definition Templates

As seen above, the complexity of the algorithm will depend on the logic of the application and the rules and regulations that enterprise abide by to secure sensitive columns. This becomes further complicated in packaged applications such as E-Business Suite, PeopleSoft and Fusion Applications. To ease this complexity, Oracle has released, E-Business Suite, Fusion Applications and will be releasing (PeopleSoft) Data Masking definition templates that work in conjunction with Application Data Model templates as described above. These data masking definition templates contain pre-defined industry best-practice masking algorithms to ensure the optimal techniques is used to securely mask the data while maintaining the application integrity to allow for correct application behavior.

With respect to E-Business Suite Data Masking template, enterprise should reference the MyOracle Support Note: 1481916.1, which provides an overview of data masking in Oracle E-Business Suite, along with instructions on how to set up the Oracle E-Business Suite Release 12.1.3 Template for Data Masking Pack with Oracle Enterprise Manager 12c Data Masking Tool.

With respect to Oracle Fusion Applications and Data Masking, you should refer to Oracle Fusion Applications Administrator's Guide Section 6.9 Data Masking in Oracle Fusion Applications. The section contains an introduction to Data Masking in Oracle Fusion Applications and a step-by-step instruction of installing the templates and executing.

When masking an Oracle Fusion Application that has been masked, only the Fusion database is masked. Hence it is important to note that the Enterprise Scheduler Service (ESS) job to synchronize the LDAP identity store and the Oracle Fusion Application database not be run. Doing so will rest the identity attributes in the database to their unmasked values.
You should also setup test users to perform the testing. One should not log into the test database as a real user or update a user's attribute in either the LDAP identity store or the Oracle Identity Manager database. Doing so will reset that user's attributes to their unmasked values.

Secure: High Performance Mask Execution

Now that the mask definition is compete, the security administrator can execute the masking process to replace all sensitive data. However before doing so, the administrators of the production environment will have to choose whether to clone the production environment into a restricted, fenced off location outside of production and mask or to perform At-Source masking. We will discuss both routes below.

Mask in cloned non-production environment

Oracle Enterprise Manager offers several options to clone the production database:

- Recover from backup: Using Oracle Managed Backups functionality, Oracle Enterprise manager can create a test database from an existing backup.

- Clone Live Database: Oracle Enterprise Manager can clone a live production database into any non-production environments within a few clicks. The clone database capability also provides the option to create a clone image, which can then be used for other cloning operations.

With the restricted, cloned non-production database now ready for masking, the Oracle Data Masking builds a work-list of the tables and columns chosen for masking. Other tables that are not required to be masked are not touched. Further, the tables selected for masking are processed in the optimal order to ensure that only one pass is made at any time even if there are multiple columns from that table selected for masking. Typically, the tables with the primary keys get masked first, followed by the dependent tables containing foreign keys.

Once the mask work list is ready, the Oracle Data Masking generates mapping tables for all the sensitive fields and their corresponding masked values. These are temporary tables that are created as a part of the masking process, which will be dropped once all data has been masked successfully.

Using a highly efficient data bulk mechanism, Oracle Data Masking rapidly recreates the masked replacement table based on original tables and the mapping tables and restores all the related database elements, such as indexes, constraints, grants and triggers identical to the original table. Compare this with the typical data masking process, which usually involves performing table row updates. Because rows in a table are usually scattered all over the disk, the update process is extremely inefficient because the storage systems attempts to locate rows on data file stored on extremely large disks. The bulk mechanism used by Oracle Data Masking lays down the new rows for the masked table in rapid succession on the disk. This enhanced efficiency makes the masked table available for users in a fraction of the time spent by an update-driven masking process. For large tables, Oracle Data Masking automatically invokes SQL parallelism to further speed up the masking process.

Other performance enhancements include using the NOLOGGING option when recreating the table with the masked data. Typical database operations such as row inserts or updates generate redo logs, which are used by the database to capture changes made to files. These redo logs are completely unnecessary in a data masking operation since the non-production database is not running in a production environment, requiring continuous availability and recoverability. Using the NOLOGGING option, the Oracle Data Masking bypasses the logging mechanism to further accelerate the masking process efficiently and rapidly.

At-Source Masking

Traditionally, sensitive and regulated information is obfuscated for non-production use out-side of production environment (described above), where a clone of the production database is established. This environment typically resided in an area away from production and is isolated from all users except for the administrators that run the masking on. Once complete and validated that no sensitive data is at risk, the environment is then further cloned and duplicated to end users such as developers, QA or third-party/off-shore consumers.

With the latest release of Oracle Data Masking Pack, another option is available called At-Source Masking. This method works by masking as the data is extracted from production database without affecting production data into masked export dump files. These masked export files can then be directly shared with end-users without the risk of exposing sensitive data in another environment outside of production. Additionally, this method provides the capability of quick mask that allows large size data to be with nulled, replaced with a fixed string or a fixed number to reduce the size of the masked export files. For example it is common in production databases to have columns that contain notes, documents and/or images. Instead of masking these using an algorithm defined in the masking definition enterprises can choose to replace them from the masked export files.

Similar to masking in a cloned environment, At-Source masking capitalizes on the optimized performance methods without affecting production data.

Masking performance tests

To assess the above optimizations in Oracle Data Masking, internal tests were conducted on an Exadata X2-2 full rack with high performance discs, 2x six core Intel ® Xeon ® X5675 processors (3.07 GHz)

TABLE 2. ORACLE DATA MASKING PERFORMANCE TESTS

DATABASE SIZE	MASK CRITERIA	EXECUTION DURATION
100GB DB	600 million rows - One column replaced	4 hrs 2 mins

	with random numbers	
1 TB DB	6 billion row table - One column replaced with random numbers	12 hrs 49 seconds
100 TB DB	600 billion row table – One column replaced with fixed number	33 mins

As these results clearly indicate, Oracle Data Masking can handle significant columns of sensitive data effortlessly.

Additionally Oracle Data Masking leverages key capabilities within the Oracle database 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 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.

Test: Integrated Testing with Application Quality Management solutions

The final step of the masking process is to test that the application is performing successfully after the masking process has completed.

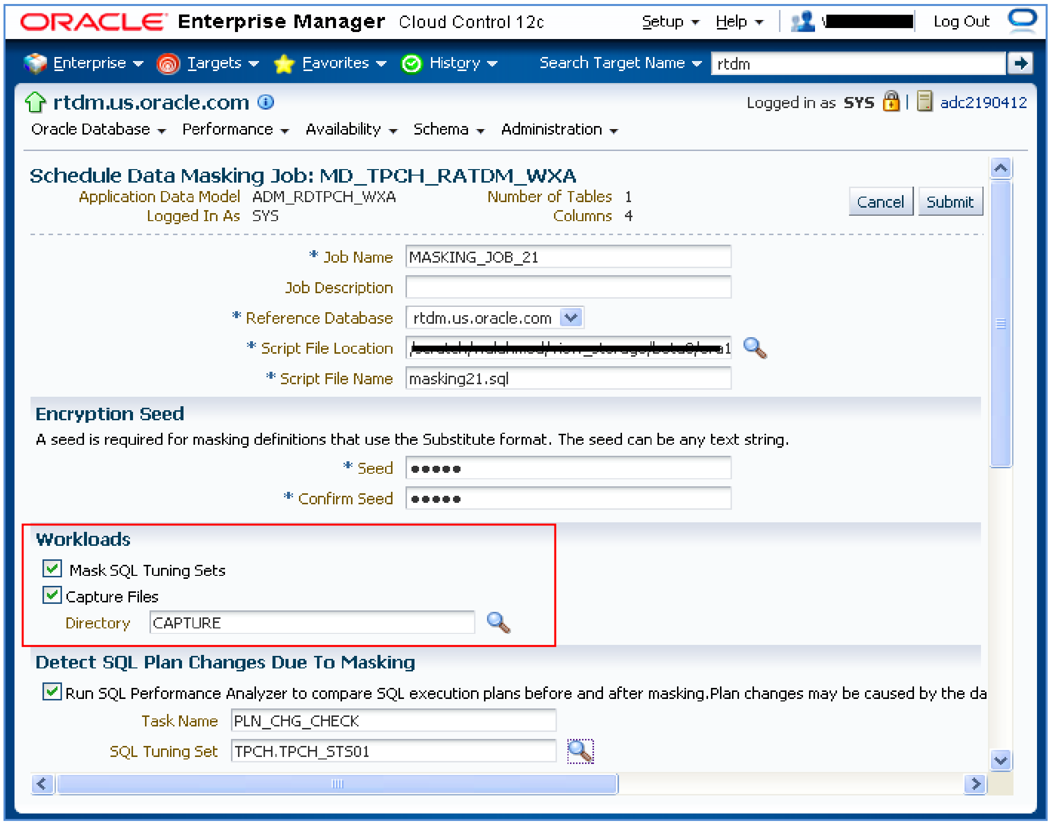
Oracle Enterprise Manager's Application Quality Management (AQM) solutions provide high quality testing for all tiers of the application stack. Thorough testing can help you identify application quality and performance issues prior to deployment. Testing is one of the most challenging and time consuming 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 you to

- Test infrastructure changes: Real Application Testing is designed and optimized for testing database tier infrastructure changes using real application workloads captured in production 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.

With respect to Real Application Testing, it is common for customers to capture production workload for the purpose of replaying in a test environment. However with stringent regulations, the possibility of this workload containing sensitive data and having the test environment masked, testing using Real Application Testing workload on masked databases was not possible until recently.

Oracle Data Masking functionality has been enhanced to work consistently across all workload artifacts. Sensitive data in database, SQL Tuning Sets and Database Replay workload capture files are masked uniformly according to the definitions specified by business and regulatory requirements defined in the Data Masking definitions. This allows for a capture performed in production to be replayed securely in a masked non-production environment compliant to data privacy regulations



Schedule Data Masking Job: MD_TPCH_RATDM_WXA

Application Data Model ADM_RDTPCH_WXA
Logged In As SYS
Number of Tables 1
Columns 4

* Job Name	MASKING_JOB_21
Job Description	
* Reference Database	rtdm.us.oracle.com
* Script File Location	[redacted]
* Script File Name	masking21.sql

Encryption Seed

A seed is required for masking definitions that use the Substitute format. The seed can be any text string.

* Seed	●●●●●
* Confirm Seed	●●●●●

Workloads

☑ Mask SQL Tuning Sets

☑ Capture Files

Directory CAPTURE

Detect SQL Plan Changes Due To Masking

☑ Run SQL Performance Analyzer to compare SQL execution plans before and after masking.Plan changes may be caused by the da

Task Name	PLN_CHG_CHECK
SQL Tuning Set	TPCH.TPCH_STS01

Figure 6. Real Application Testing Integration with Data Masking

In addition to this enhancement, Oracle Data Masking Pack allows customers to run SQL Performance Analyzer as part of the masking process. This allows customers to compare execution plans before and after masking to anticipate application performance issues in non-production environments.

Data Masking and Subsetting Integration

Data stores are expanding at a rate greater than ever and the number of database applications increasing, enterprises are faced with the increased provisioning of non-production environments for the purpose of application development and testing purposes as mentioned earlier. However, they cannot afford to incur the storage expenses of provisioning the same production data in their non-production environments, nor do they have the tools or the application knowledge to shrink the data to a right-sized non-production environment.

Oracle Test Data Management Pack helps enterprises to shrink storage costs by creating a reduced size copy of production data for non-production use (testing, development, training) while maintaining the referential integrity of the data set.

Integration of Data Masking and Test Data Management, allows enterprise to use a simple workflow, to reduce the size of the database simultaneous with the masking of sensitive data. This serves the dual purpose of sanitizing the sensitive exported data while greatly reducing the hardware cost related to storing large masked production database for testing.

As part of the integration between Data Masking and subsetting, enterprises can, as part of the process of securely subsetting the production data significantly reduce the subset size by discarding columns containing large chunks of data by defining column rules to set CLOB and BLOB to null (or another supported format such as Fixed String, Fixed Number).

Additionally, with this capability enterprises can mask or subset and mask at the source (as described above). This ensures that sensitive data never leaves the source database when provisioning test systems and therefore comply with data protection policies.

This integration allows for the best practice of efficiently masking only the required sensitive data, while maintaining a referentially intact subset application, reducing the overhead of storage, complex manual tasks and increasing robust automation in the growing provisioning requirement of secure non-production databases.

Support for heterogeneous databases

The reality in today’s enterprises is that data resides in systems that are not Oracle databases and are held to the same standards of regulatory requirements.

Oracle Data Masking supports masking of sensitive data in heterogeneous databases such as IBM DB2, Microsoft SQL Server through the use of Oracle Database Gateways and will follow the process as:

1. Production data copied to test
2. Sensitive data copied to staging database
3. Sensitive data masked in staging

Masked data copied from staging to test

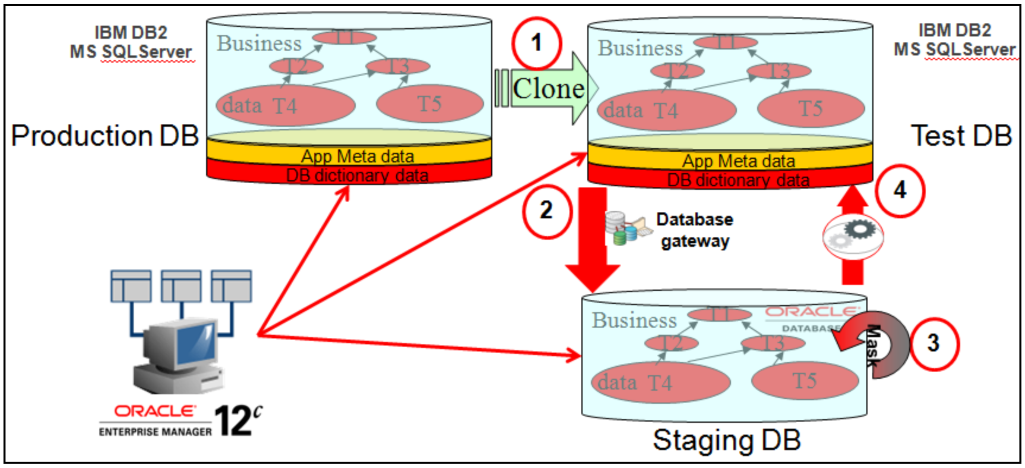
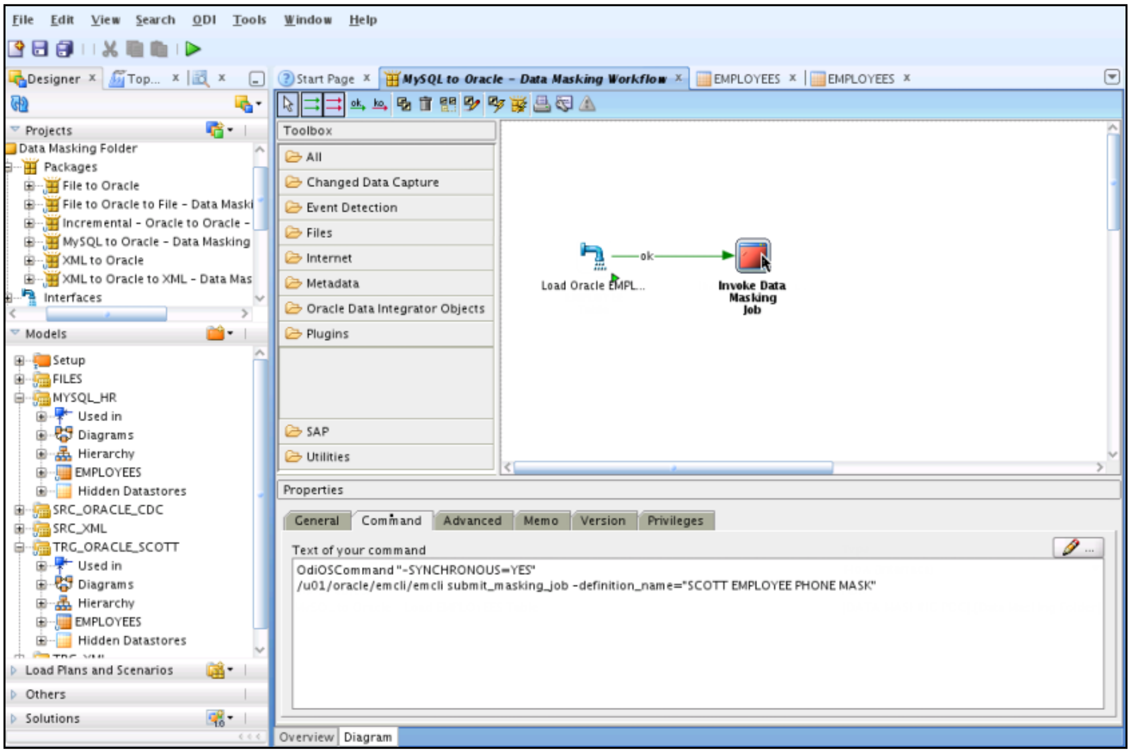


Figure 7. Heterogeneous Data Masking

Additionally, sensitive data may reside in operating system flat file, XML documents or mySQL. Oracle Data Integrator can extract this sensitive information, mask by calling Oracle Data Masking, all in a unified workflow.



Text of your command

```
OdiOSCommand "-SYNCHRONOUS=YES"
/u01/oracle/emcli/emcli submit_masking_job -definition_name="SCOTT EMPLOYEE PHONE MASK"
```

Figure 8. Oracle Data Integrator and Data Masking integration

Customer Case Studies

Customers have a variety of business needs which drive their decision to adopt Oracle Data Masking for their sensitive enterprise data. We will discuss some of the many customers taking advantage of these benefits to quickly realize the return on investment.

A major global telecommunications products company that implemented the above methodology understood the challenges that they were facing and reaped the benefits of Oracle Data Masking. 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, 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.

Financial companies are strictly held accountable for ensuring sensitive information simply by the nature of the business. This, culminated with the aggressive climate to stay competitive, a leading financial institution based out of the US decided to implement Oracle Data Masking. With the challenge of exploding data volume and the reduced release cycle to just two weeks, development activities required usable environments as close to production as possible without sacrificing security of their customer data to promote reliable changes into production. As a result of this business demand and legal requirements, the legacy masking tool that performed serially, supporting longer development cycle become inadequate and needed to be replaced. The business needed a solution that would mask sensitive information, retain referential integrity and execute in a fraction of the time while not adding significantly to the DBA workload. The financial institution was able to install and use Oracle Data Masking in a matter of just two weeks. The DBA’s were able to then start provisioning secure development environments in a fraction of the time (96% less) of their legacy system. This met the aggressive release cycle for the business to stay competitive in the market while ensuring compliance with regulations.

Having a standardized method to mask all data in an enterprise results in a solution that can be scalable and flexible to future demands. A global communications and information technology company with several other diversified product offerings, struggled with the challenges of not having a standardized tool where confidential information is masked in non-production environments. They needed a mechanism by which to eliminate manual analysis and efforts, to manage duplicate non-production environments containing sensitive data. The solution needed to ensure compliance of worldwide Data Privacy rules and regulations without hampering proper functioning of applications. Following the analysis of data masking solution vendors, the company selected Oracle Data Masking as the clear winner. Since Oracle Data Masking is integrated into Oracle Enterprise Manager which is used to manage the environments databases, the learning curve is significantly reduced. Additionally, with the out-of-the-box pre-defined masking algorithms, masking common sensitive information, such as social security numbers and credit card numbers reduced significant time in constructing from scratch. With user access control and a change control process in the solution, the company was able to ensure that the sensitive data was masked in the required methods by the right individuals.

Conclusion

Staying compliant with policy and government regulations while sharing production data with non-production users has become a critical business imperative for all enterprises. Oracle Data Masking is designed and optimized for today’s high volume enterprise applications running on Oracle databases. Leveraging the power of Oracle Enterprise Manger to manage all enterprise databases and systems, Oracle Data Masking accelerates sensitive data identification and executes the masking process with a simple easy-to-use web interface that puts the power of masking in the hands of business users and administrators.

Additionally, the flexibility of Oracle Data Masking allows enterprise to face the rapid changing climate of the future by providing robust out-of-the-box algorithms, pre-defined templates and customizable definitions optimized to run in a fraction of the time of traditional masking methods.

Organizations have benefited by implementing Oracle Data Masking to protect sensitive information in non-production environments to comply with regulatory requirements such as

Sarbanes-Oxley, Payment Card Industry Data Security Standard (PCI DSS), Health Insurance Portability and Accountability Act of 1996 (HIPAA) as well as numerous other laws and regulations that restrict the use of actual customer data. The benefits include amongst many:

- **Reducing Risk through 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.

- **Increasing Productivity through 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.

ORACLE®

White Paper Title
June 2013
Author: Waleed Ahmed, Jagan Athreya
Contributing Authors:

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

Hardware and Software, Engineered to Work Together