

---

## Example - Database Masking (In-Place) Exercise

- Please feel free to use your own masking engine
- For this exercise/example I used my masking engine  
<http://md5350dc4.dc4.delphix.com/masking/>
- Prepare a target host and database or db, example below:
  - 10.43.16.18 ([mdora11204.dcenter.delphix.com](http://mdora11204.dcenter.delphix.com))  
o.s. logon = oracle/oracle
- target database information, example:
  - sid: example VDBTEST1

db schema to be masked: SCOTT

db masking login user: SCOTT password: scott

Note: the db masking user does not need to be the db schema owner. The requirement is that the db masking user should have the correct update privileges to the table(s) to be masked.

In this example we are using a table , table description and test data values as shown in this screenshot:

```
SQL> desc employee_details
Name                                     Null?    Type
-----
EMP_NUMBER                             NOT NULL NUMBER(8)
FIRST_NAME                             VARCHAR2(10)
LAST_NAME                              VARCHAR2(15)
BACNK_ACCOUNT_NUMBER                   NUMBER(8)
DEPARTMENT1                           NUMBER(3)

SQL> select * from employee_details
2 ;
```

EMP_NUMBER	FIRST_NAME	LAST_NAME	BACNK_ACCOUNT_NUMBER	DEPARTMENT1
10000001	Mary	Christoff	99007321	10
10000002	CRAIG	ALDER	23451234	5
10000003	SANDRO	Lentini	45679900	3
10000004	PHILLIP	WEEDON	32321234	20
10000005	Sofia	Al-ghabban	12347890	15

To check what the data looked like pre- and post-masking, example:

- i) `ssh oracle@10.43.16.18`
- ii) `export ORACLE_SID= VDBTEST1`
- iii) Connect via sqlplus, e.g.

`sqlplus scott/scott`

To see the table definition and table data, issue:

```
desc employee_details
select * from employee_details
```

Let's try masking!

- 1) Login to the Masking Engine, in this example:

<http://md5350dc4.dc4.delphix.com/masking/>

login=admin/Admin-12

- 2) Provide the Masking engine the details that it needs to be able to connect and access the target data or data to be masked

- i) Click on the Environments tab, then "Add Application".
- ii) Create an 'Application' tag - under which Environments can be created.

For example: "HR" or "Personnel" , with the title normally representing what type of application uses the data that is to be masked

- iii) Follow with "Add Environment". Environment examples could be 'UAT'/'Production'/'Development'/'Final\_testing'/'Training'

Note: Connectors/RuleSets/Inventory/Jobs are created under an Environment.

iv) Create a Connector to be used to access the target data

Click on Environments -> select your Environment from the List

Click on Connector Tab —> Create Connection

Connection details, example:

Type: Database - Oracle

Connection name:

Schema Name: SCOTT

Database name: VDBTest`

HostName/IP: 10.43.16.18

Port: 1521

Db masking user login Id: SCOTT <— - this user need to have update privileges on  
table(s) to be masked

Password: scott <— — please check that you can connect to target db with the  
db user id and password that you provide

Note: the details above are examples for an Oracle target database.

Test the Connection, then Save if connection successful.

v) Create a RuleSet

Click on 'Rule Set' Tab —> Create Rule Set

Use/select the connector previously created

Select the table(s) to be masked and click on 'Save'.

Note: A ruleset can be made up hundreds of tables and columns.

vi) Check table columns list via the Inventory

Environments -> Inventory -> then select your RuleSet from ruleset list

This will list the columns for each table in the ruleset

- 3) Define or create a masking algorithm that will be applied to the target data
- algorithms are independent of applications and environments and can be used on both database and file masking jobs

Click on Settings -> Algorithms -> Add Algorithm

- a) Create a Segment Mapping Algorithm, provide details as below

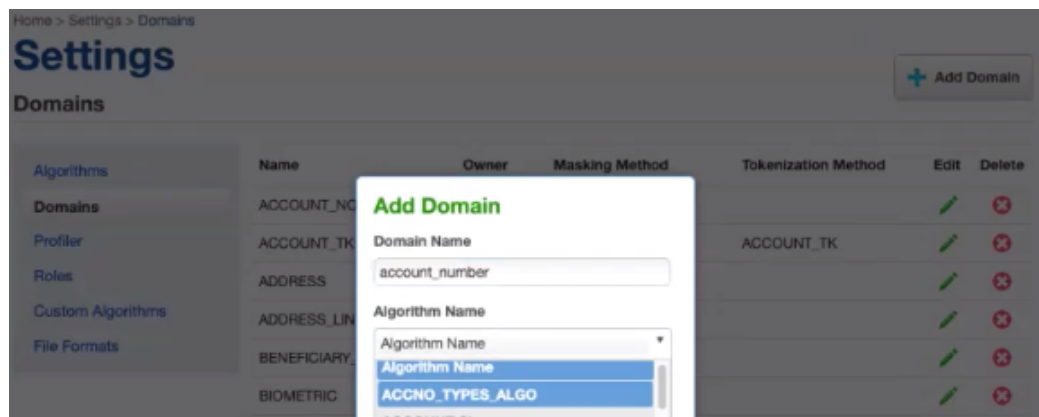
The screenshot shows the configuration for a Segment Mapping Algorithm. On the left is a sidebar with a list of algorithm types: Secure Lookup Algorithm, Segment Mapping Algorithm (selected), Mapping Algorithm, Binary Lookup Algorithm, Tokenization Algorithm, Min Max Algorithm, Data Cleansing Algorithm, and Free Text Redaction Algorithm. The main panel is titled 'Algorithm Name' and contains the text 'accno\_types\_algo'. Below this is the 'Description' field, which contains 'masking algo for account number type columns - 8 numbers-wide'. The 'Number of Segments' is set to 2. The configuration is divided into two sections, 'Segment 1' and 'Segment 2'. Each segment has a 'Numeric' dropdown and a '4' value field. Below these are two columns of input fields: 'Real Values' and 'Mask Values'. Each column has three sub-headers: 'Min #', 'Max #', and 'Range #', each followed by an empty input box.

Algorithm Configuration								
Algorithm Name			accno_types_algo					
Description			masking algo for account number type columns - 8 numbers-wide					
Number of Segments			2					
Segment 1								
Numeric			4					
Real Values			Mask Values					
Min #	Max #	Range #	Min #	Max #	Range #			
Segment 2								
Numeric			4					
Real Values			Mask Values					
Min #	Max #	Range #	Min #	Max #	Range #			

- b) Define a Domain and assign the appropriate newly created algorithm to it

Domain is a way of giving a name for the type of data being masked by a particular algorithm. For example a domain name= ACCOUNT\_NUMBER, possible algorithm to be used = Account\_SL

Click on Settings -> Algorithms -> Add Domain



4) Assign masking algorithms to the columns to be masked

Environments -> Inventory -> then select your RuleSet from ruleset list

This is the list of tables and the candidate columns for masking, and here you can assign appropriate algorithm to the columns to be masked. Our example has only one table in the ruleset.

Click on the Green Pencil Icon (Edit icon) against the following columns and assign the algorithms as in the following screenshots and click on Save.

LAST\_NAME, BACNK\_ACCOUNT\_NUMBER

The screenshot shows the 'Edit Properties' form. It has two columns of fields. The left column contains: 'Column Name' (LAST\_NAME), 'Data Type: VARCHAR2 (15)', 'Domain' (LAST\_NAME), and 'Algorithm' (LAST NAME SL). The right column contains: 'ID Method' (Auto) and a 'Notes' text area. At the bottom are 'Cancel' and 'Save' buttons.

## Edit Properties

Column Name	ID Method
BACNK_ACCOUNT_NUMBER	Auto
Data Type: NUMBER (22)	Notes
Domain	
account_number	
Algorithm	
ACCNO_TYPES_ALGO	

Cancel Save

5) You are now ready to create and run a masking job

Click on 'Environments' tab, click on your new Environment, click on 'Mask'

- Provide Job Name
- Masking Method: 'In Place'
- Select the RuleSet you created from the 'Rule Set' list
- Leave all the fields at default

## Create Masking Job

Job Name	Commit Size	Feedback Size
hr_masking_job1		
Masking Method	<input type="checkbox"/> Disable Trigger	
In-Place	<input checked="" type="checkbox"/> Batch Update <input type="checkbox"/> Disable Constraint	
Target: HUMAN_RESOURCES	Prescript	
<input type="checkbox"/> Multi Tenant	Select...	
Rule Set	Postscript	
Rule Set	Select...	
md_hr_ruleset1	Comments	
Min Memory	Max Memory	Email
In MB	In MB	marisa.damaso@deiphix.com
Update Threads		
1		
If Nonconforming Data is encountered		
<input type="checkbox"/> Stop job on first occurrence		

Cancel Save

Save, then run the masking job by clicking on the Run Job icon under 'Action'

Home > Environments > HUMAN\_RESOURCES

## HUMAN\_RESOURCES

[Profile](#) [Mask](#)

**Environment**

**Name** HUMAN\_RESOURCES

**Purpose** Mask

**Application Name** User\_acceptance...

**Approval workflow** Disabled

Job ID ▾	Name	Rule Set	Completed	Status	Action	Edit	Delete
9	hr_masking_job1	md_hr_ruleset1	...	Created			

A successful masking job run:

Home > Environments > HUMAN\_RESOURCES

## HUMAN\_RESOURCES

[Profile](#) [Mask](#)

**Environment**

**Name** HUMAN\_RESOURCES

**Purpose** Mask

**Application Name** User\_acceptance...

**Approval workflow** Disabled

Job ID ▾	Name	Rule Set	Completed	Status	Action	Edit	Delete
9	hr_masking_job1	md_hr_ruleset1	2020-03-27 15:58	Succeeded			

To see further information about that job run or execution, click on the highlighted job name, and this takes you to the Monitor page as below:

Environments

Monitor

Scheduler

Settings

Admin

Audit

Home > Monitor > Completed

Monitor

0

Jobs Running

hr\_masking\_job1

100%

★ SUCCESS

Environment	Start Time	15:58:36	Total Time Taken	00:00:05
HUMAN_RESOURCES	Previous Run Time		Masking Report	<a href="#">Download Report</a>
Job ID	Total # of Tables	1	Masking Inventory Report	<a href="#">Download Report</a>
9	Tables Masked	1	Rows Remaining	0
Execution ID	Tables with Nonconforming Data	0	Rows Masked	5
60	Tables to be Masked	0	Columns with Nonconforming Data	0
CM Connection	Job Type	Mask	Streams	1
table			Updates Running	1
Source / Target			Repository	POSTGRESQL
- / SCOTT				

Completed

Processing

Waiting

Completed

1

Complete

1

Total Tables

Name	Progress	Time	Rows Per Min	Rows Masked	Rows Remaining	Columns with Nonconforming Data
EMPLOYEE_DETAILS	100%	★	0d 0h 0m	145	5	0

6) Check the masked data. In our example we can see that the LAST\_NAME and BACNK\_ACCOUNT\_NUMBER columns had been masked:

```
SQL> select * from employee_details;
```

EMP_NUMBER	FIRST_NAME	LAST_NAME	BACNK_ACCOUNT_NUMBER	DEPARTMENT1
10000001	Mary	Reed	54923357	10
10000002	CRAIG	Theobald	53852240	5
10000003	SANDRO	Lawley	78263744	3
10000004	PHILLIP	Carwin	86092240	20
10000005	Sofia	Craig	5977049	15