



DATPROF

Feature Review Deterministic Masking

19 March 2020

DATPROF – Test Data Simplified

Introduction

DATPROF Privacy Version 4.1 delivered in March 2020 and delivers two new key features to our data masking toolbox:

- Support for Deterministic Masking when using generated data
- Support for the MySQL and MariaDB database platforms bringing the platform list to:
 - Oracle
 - Microsoft SQL Server
 - IBM DB2 for Linux, Unix and Windows
 - IBM DB2 for iSeries
 - PostgreSQL + EnterpriseDB
 - MySQL + MariaDB

This paper focuses upon Deterministic Masking which has been implemented across all supported platforms.

What is Deterministic Masking?

Deterministic Masking is the process of replacing a value in a column with the same value whether in the same row, the same table, the same database/schema and between instances/servers/database types.

Example: A database has multiple tables, each with a column that has first names. With deterministic masking the first name will always be replaced with the same value - "Lynne" will always become "Denise" – wherever "Lynne" may be in the database.

What's the benefit of Deterministic Masking?

In a nutshell the benefit is consistency of the masked output within and across masking projects regardless of the database or, indeed, the RDBMS platform.

Every downstream database user tends to have a favourite account he or she works with. After a database refresh the masked data values would have historically been different between refreshes. Now, it will be the same.

QA and Test teams want to have data consistency so that their manual or automated procedures can be reliably and consistently deployed.

I thought I could do that with Privacy anyway?

In previous versions of DATPROF Privacy it was possible to achieve Deterministic Masking with the use of Translation Tables. These tables are generated afresh during each masking run and subsequently used within the run as “lookup tables” for masking functions. This achieved a quasi-Deterministic Masking within the current masking project but the next time the project is run the results would be different (although consistently applied).

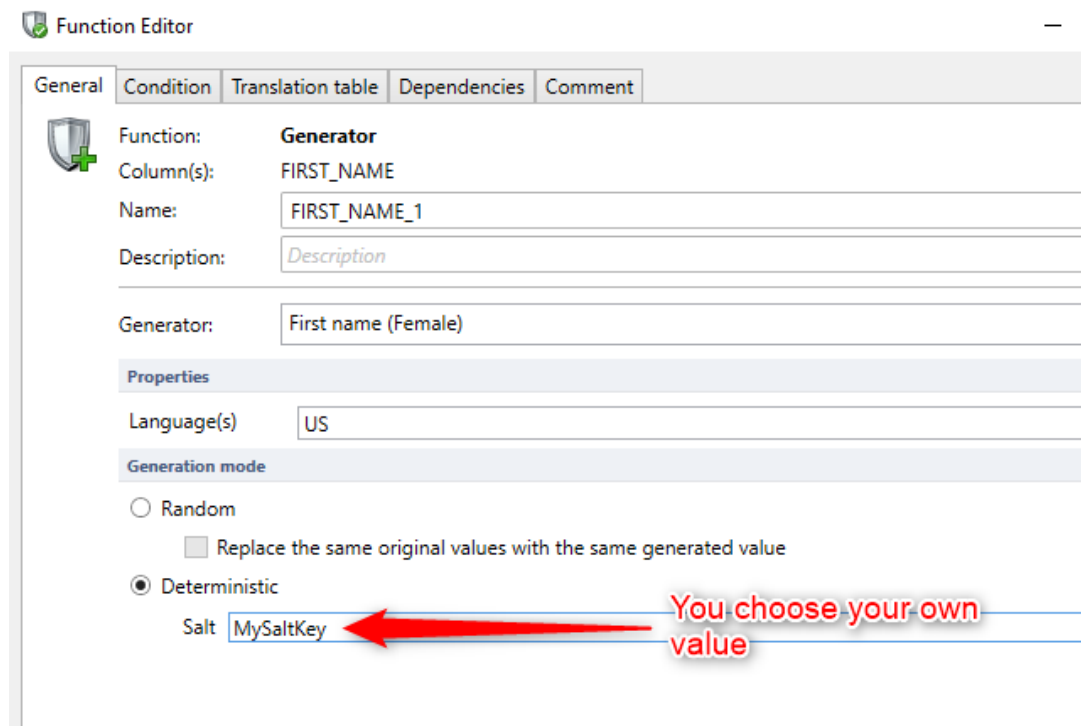
Implementing a practice where the Translation Tables are secured and re-used for lookup would achieve consistency between runs but requires actions to be specified in the event that a value to be masked does not have a corresponding value in the lookup table.

Translation Tables, by their nature, provide a way of reverse engineering the data masking process should they be exposed. By implication they needed to be held in a secure manner and transposed across instances / servers to achieve consistent masking across the Estate.

Our new Deterministic Masking feature removes all of these considerations.

How does it work?

In concept, the replacement of a data value with a generated alternative is determined by a user specified “Salt” key. This is “combined” with the replacement datasets themselves to determine which row in the replacement dataset is chosen.

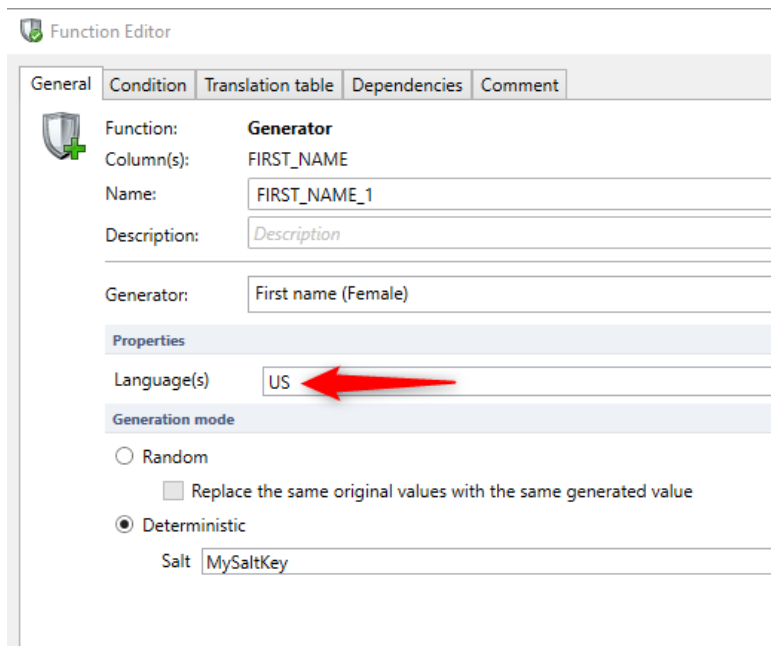


The screenshot shows the 'Function Editor' window with the following configuration:

- Function: **Generator**
- Column(s): FIRST_NAME
- Name: FIRST_NAME_1
- Description: *Description*
- Generator: First name (Female)
- Language(s): US
- Generation mode:
 - Random
 - Replace the same original values with the same generated value
 - Deterministic
 - Salt: MySaltKey

A red arrow points from the text "You choose your own value" to the "Salt" field containing "MySaltKey".

The choice of the replacement Properties...



Function Editor

General | Condition | Translation table | Dependencies | Comment

Function: **Generator**


Column(s): FIRST_NAME

Name: FIRST_NAME_1

Description: *Description*

Generator: First name (Female)

Properties

Language(s) **US** 

Generation mode

Random

Replace the same original values with the same generated value

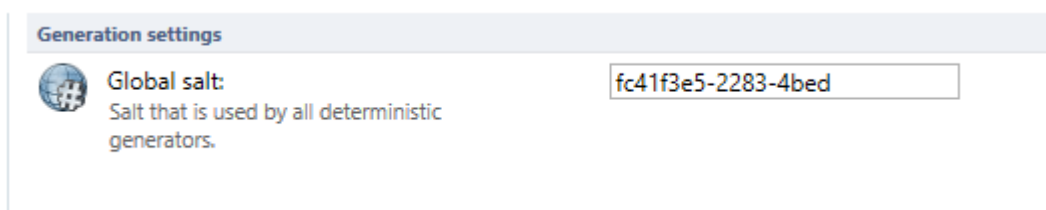
Deterministic

Salt MySaltKey


...has a significant impact on the replacement candidate row selection.

Remember, you can select any combination of properties to achieve your masking objectives but to achieve consistently masked values **you must ensure that the replacement property selection is consistent between tables and databases/schemas (projects)**.

There is also a concept of a Project Global Key which is the over-arching value in the replacement value selection algorithm. This value Privacy generated value can be found in the Deployment tab under Settings:



Generation settings

 **Global salt:**

Salt that is used by all deterministic generators.

The Global salt value is generated when the project is created or upgraded from a previous version of Privacy. It is, however, fully editable so you can replace it with a Global key created in another project.

It follows, therefore, that using a deterministic technique will allow you to implement consistent data masking across your estate if you:

- Use the same Global salt key across all projects
- Use the same Deterministic salt value at the rule level and
- Keep the Properties the same for the Generator which you select in a rule

Conclusion

This feature reinforces DATPROF Privacy's position as a Global Leader in the provision of test data masking solutions.

For more information or to arrange a personal demonstration of DATPROF Privacy please contact us at any time.

About DATPROF

Behind all products and services there is a team that builds and supports it. From phones and cars, to bank accounts and health insurance, there are developers and testers working across the world that rely on good quality test data.

It is for these people that we build our products to enable them to focus on what's really important by giving them the right data to innovate faster and ship their products with higher quality. In the end everyone benefits from these innovations.

We **simplify** getting the **right test data** to the **right place** at the **right time!**



DATPROF – Test Data Simplified

Telephone: +31 (0)505710305

Email: Info@datprof.com

Website: www.datprof.com

Copyright © DATPROF

This publication and no part of it may be reproduced without prior written permission from the publisher. Also placing direct links to the file location of this document is not permitted.