

**Optim Data Privacy Functions for Data Masking**

Auteurs: Knights Harnasses team

Version: 0.91

Date: juli 2020

# Documentbeheer

**Versiebeheer**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versie** | **Datum** | **Auteur** | **Wijzigingen** |
| 0.1 | 05-7-2020 | Ganesh Nikumbh | Initiele versie |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Distributielijst**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versie** | **Naam afdeling** | **Naam contactpersoon** | **Opmerkingen** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Er is afgestemd (advies/overleg) geweest met de volgende partijen:

|  |  |  |
| --- | --- | --- |
| **Naam afdeling** | **Naam contactpersoon** | **Opmerkingen** |
|  |  |  |
|  |  |  |

Inhoud

[Documentbeheer 3](#_Toc45053918)

[1 Algemene beschrijving 5](#_Toc45053919)

[1.1 Introductie 5](#_Toc45053920)

[1.2 Scope / afbakening 5](#_Toc45053921)

[2 Optim User defined functions 5](#_Toc45053922)

[6](#_Toc45053923)

[3 Optim Data Privacy provider library 7](#_Toc45053924)

[3.1 Affinity Privacy Provider (pro=aff) 7](#_Toc45053925)

[3.2 Age Privacy provider 9](#_Toc45053926)

[3.3 Email Privacy Provider 10](#_Toc45053927)

# Algemene beschrijving

## Introductie

This document describes the Optim data privacy solution user-defined functions functions (UDFs) and their examples, which can be useful to mask personal and sensitive data in DIM datawarehouse. The optim data privacy UDFs use the optim data privacy providers (**odpp**) libraries, installed standlone on oracle database.

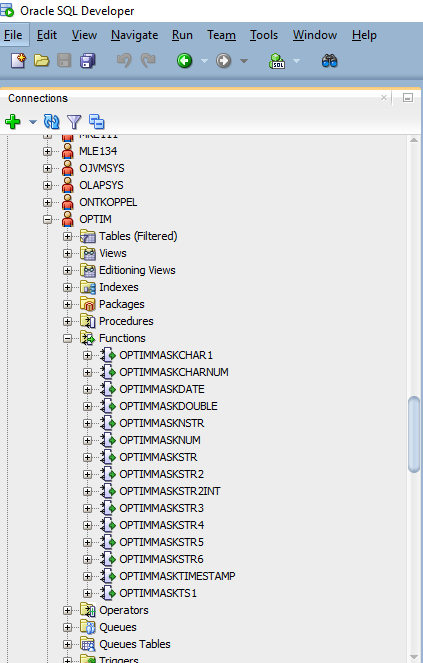
## Scope / afbakening

This document describes only some of the optim UDFs and data privacy providers which are most commonly used for data masking, and can be useful for Data masking requirements in DIM. For additional information on optim function refer to below resource link

<https://www.ibm.com/support/knowledgecenter/en/SSMLQ4_11.3.0/com.ibm.nex.optimd.tdm.doc/topics/optdm-c-converting_data.html>

# Optim User defined functions

Optim user defined funtions (UDFs) are installed on DIM oracle database, in a separate schema. The schema name is optim.



Below are the most common used UDFs and their syntax

1. OptimMaskStr

Syntax :

OptimMaskStr(*input\_argument* (datatype=varchar), *ParamString* (parameter string) ) -> ouput (datatype=varchar)

*e.g. select OptimMaskStr('340042404908916','pro=ccn,flddef1=(name=c1,dt=char)') from dual;*

1. OptimMaskDate

Syntax :

OptimMaskDate(*input\_argument* (datatype=date), *ParamString* (parameter string) ) -> ouput (datatype=date)

*e.g. select OptimMaskDate('02-MAR-76', 'pro=age, mon=1,yr=1,flddef1=(name=dt1,dt=date)') as maskdt1 from dual;*

1. OptimMaskNum

Syntax :

OptimMaskNum(*input\_argument* (datatype=number), *ParamString* (parameter string) ) -> ouput (datatype=number)

*e.g. select OptimMaskNum(340042404908916,'pro=ccn,flddef1=(name=c1,dt=u\_long\_long)') as masked from dual;*

In above funtions:

* *input\_argument* is the input value which need to be masked
* *ParamString* is the string of parameters which describes the provider library need to be used for masking along additional parameters (if any)

*e.g. incase of credit card number masking, the ParamString can be given as*

*‘pro=ccn, mtd=rep, wheninv=pre, flddef1=(name=ccnchar, dt=char)’*

*Here,*

*pro= provider (ccn=credit card number). This is a mandatory parameter in ParamString for all masking providers.*

*flddef= input field definition. This is a mandatory parameter in ParamString for all masking providers.*

*Details about the mandatory and additional parameters like mth (method), wheninv (when invalid values found at source) can be found in* [*here*](https://www.ibm.com/support/knowledgecenter/SSMLQ4_11.3.0/com.ibm.nex.optimd.tdm.doc/oxy_ex-1/com.ibm.nex.mod.doc/topics/opmod-c-optim_data_privacy_providers.html) *or in below section.*

# Optim Data Privacy provider library

The Optim data privacy provider (odpp) library, is a stand-alone API that provides a flexible and extensible means of accessing predefined data masking providers. The library includes a set of data privacy algorithms that are called data privacy providers, which are sometimes referred to as data masking providers. Various providers are available to mask data. These include providers that are designed to mask credit card numbers, national IDs (for few countries), email addresses, birth dates and other dates, and undifferentiated or dynamically formatted values.

The library also includes providers that feature various masking techniques:

* Use the affinity privacy provider to mask data while maintaining the format and character types of the source values.
* Use the data swapping privacy providers to “swap” data within a selected field of source data.
* Use the hash privacy provider to mask source data with numeric values generated by a hash algorithm.

## Affinity Privacy Provider (pro=aff)

Use the affinity privacy provider to mask data while maintaining the format and character types of the source values, so the data type remains unchanged.

The provider masks alphabetic and numeric characters, but copies other characters in the source data to the destination.

For example, a credit card number in the format *"nnnn nnnn nnnn nnnn"* is masked to a different number that includes spaces at the same intervals, while one formatted with dashes is masked as *"nnnn-nnnn-nnnn-nnnn"*.

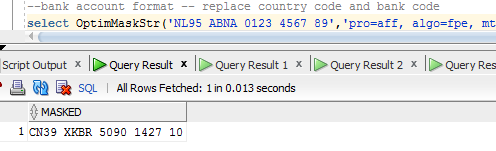
The provider includes two algorithms for masking data, a default algorithm and a format preserving encryption (FPE) algorithm.

The FPE algorithm is encryption-based and offers stronger masking capability than the default. The FPE algorithm is based on the Advanced Encryption Standard 256-bit (AES-256) algorithm, can optionally use an encryption key that is supplied by the user, and can produce masked values that are unique. The same user-supplied key produces repeatable masked values. For this reason, knowledge of the key should be secured from unauthorized users to prevent reverse engineering to discover the original values.

Example:

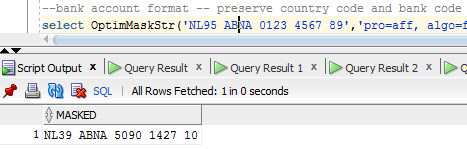
1. Masking of account number 'NL95 ABNA 0123 4567 89' with format preserving encryption

select OptimMaskStr('NL95 ABNA 0123 4567 89','pro=aff, algo=fpe, mtd=rep, key="Xyz12",  rule=num, whenmatch=shuffle, flddef1=(name=acctnbr, datatype=varchar datatype=varchar)') as masked from dual;



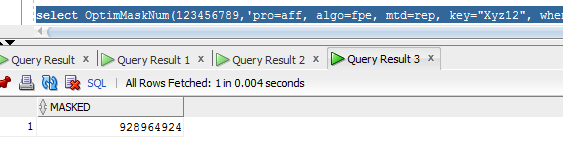
1. Masking of account number 'NL95 ABNA 0123 4567 89' with format preserving encryption(fpe) and copy the country code and bank id from source

select OptimMaskStr('NL95 ABNA 0123 4567 89','pro=aff, algo=fpe, mtd=rep, key="Xyz12",COPY = (1,2)(6,4),  whenmatch=shuffle, flddef1=(name=acctnbr, datatype=varchar datatype=varchar)') as masked from dual;



1. Masking a number with fpe

select OptimMaskNum(123456789,'pro=aff, algo=fpe, mtd=rep, key="Xyz12", whenmatch=shuffle, flddef1=(name=bsnnbr, datatype=int)') as masked from dual;



Above example uses the following parameters:

pro=aff

Provider library used is aff (affinity)

algo=fpe

The FPE algorithm offers a high level of encryption and produces masked values that are much less likely to reveal the original source value than the default.

mtd=rep

This parameter produces consistent target values when the same data is processed multiple times.

key=”User provided secret key”

This parameter provides the case-sensitive encryption key for the FPE algorithm. To produce repeatable results, use the same key value.

rule=num

This parameter ensures that numeric source values produce numeric target values.

whenmatch=shuffle

This parameter prevents identical source and target values by adding shuffling the target values that match the source.

More details about this provider and parameters can be found [here](https://www.ibm.com/support/knowledgecenter/SSMLQ4_11.3.0/com.ibm.nex.optimd.tdm.doc/oxy_ex-1/com.ibm.nex.mod.doc/topics/opmod-r-affinity_service_provider.html#affinityserviceprovider__CHARParameters).

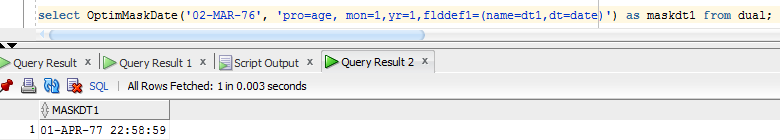
## Age Privacy provider

Use the age privacy provider to mask date values.

The aging process can increment or decrement a date value. Aging can be specific to the number of years, months, weeks, or days. Optionally, aging can be a combination of these units. Aging can also be based upon a specific 4-digit year value.

Example:

1. Below example ages the input date by 1 month and 1 year

select OptimMaskDate('02-MAR-76', 'pro=age, mon=1,yr=1,flddef1=(name=dt1,dt=date)') as maskdt1 from dual;

The different aging parameters are given below:

[Aging parameters](https://www.ibm.com/support/knowledgecenter/SSMLQ4_11.3.0/com.ibm.nex.optimd.tdm.doc/oxy_ex-1/com.ibm.nex.mod.doc/topics/opmod-r-age_service_provider.html?view=kc#ageserviceprovider__AgingParameters)

[ YEAR = *n-years* | “specific-year“ } ] ,

[ MONTH = *n-months* ] ,

[ WEEK = *n-weeks* ] ,

[ DAY = *n-days* | ANY ] ,

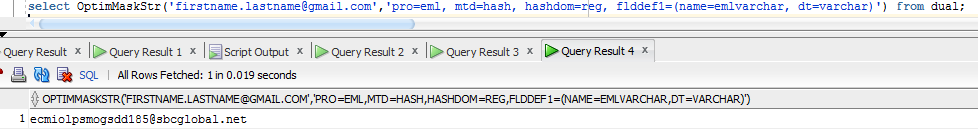
[ PIVOT = *century-pivot-value* ] ,

More details about this provider and parameters can be found [here](https://www.ibm.com/support/knowledgecenter/SSMLQ4_11.3.0/com.ibm.nex.optimd.tdm.doc/oxy_ex-1/com.ibm.nex.mod.doc/topics/opmod-r-affinity_service_provider.html#affinityserviceprovider__CHARParameters)

## Email Privacy Provider

Use the email privacy provider to generate a masked email address from source email id. An email address consists of two parts, a user name and a domain name, which is separated by '@'. For example, user@domain.com.

Example:

select OptimMaskStr('firstname.lastname@gmail.com','pro=eml, mtd=hash, hashdom=reg, flddef1=(name=emlvarchar, dt=varchar)') from dual;

Below are the parameter for this provider library.

Hash

The following example uses the hash method and a registered domain name.

pro=eml, mtd=hash, hashdom=reg, flddef1=(name=emlvarchar, dt=varchar\_szvarchar)

This example uses the following parameters:

MTD=HASH

This parameter hashes the source value to generate a user name that is composed of alphanumeric characters.

HASHDOM=REG

This parameter selects domain names from a list of large email service companies.

More details about this provider and parameters can be found [here](https://www.ibm.com/support/knowledgecenter/SSMLQ4_11.3.0/com.ibm.nex.optimd.tdm.doc/oxy_ex-1/com.ibm.nex.mod.doc/topics/opmod-r-affinity_service_provider.html#affinityserviceprovider__CHARParameters).

## Some more example

Attached sample sqls give more example about other privacy providers like ccn (credit card number) and about additional optim UDFs present in oracle database.

