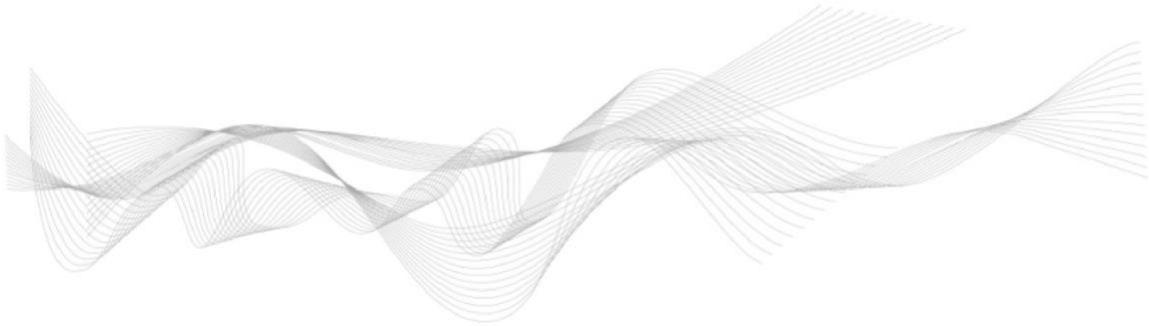


D E L P H I X



Delphix Masking Inventory Reassignment Guide

Legacy Custom Algorithm Replacement

Supporting Delphix Version 6.0.8 and above

Updated April 2022

Contents

Purpose of this document Prerequisites

Process Summary Process Detail

- I. Identify mapplets and determine replacements
- II. Create optional backup of masking inventories employing mapplet
- III. Replace mapplets in inventories with extended (new framework) algorithms
 - A. Select method for inventory update
 - B. Schedule mapplet replacement
 - C. Execute mapplet replacement
- IV. Verify mapplets are no longer assigned to masking inventories
- V. Verify Domains are updated to new algorithms
- VI. Remove mapplets from masking engine

Legacy Algorithm Framework Transition

Appendix

DXM toolkit guidance for algorithm replacement

Mapplet/Extended Name algorithm masked value comparison

Purpose of this document

This document will help guide users through the process of updating existing inventories to leverage the new Delphix algorithm frameworks using the API Client and/or DXM Toolkit. The API Client includes a user interface, is delivered with Delphix Masking and is recommended for most use cases discussed in this document. The DXM Toolkit is an open source project created by Delphix Services which uses the API to ease repetitive activities. A download link for the DXM Toolkit is found in the appendix.

Processes described in this document will facilitate transition from the legacy algorithm frameworks to the new frameworks which will be required prior to upgrading to masking engine versions following removal of the legacy frameworks planned for July 2022. Notification of deprecation of Legacy Custom Algorithms (aka mapplets) was published in January 2021 and is found [here](#). During this transition phase all OOB algorithms will also be transitioned to the extended algorithm frameworks (see Legacy Algorithm Framework Transition section later in this document for more info). Only the legacy custom algorithms (mapplets) will require inventory changes which is the focus of this document.

Many Delphix users have installed legacy custom algorithms to meet specific use cases, many of which are now met with out-of-the-box (OOB) algorithms. The most common use cases for mapplets included case sensitive name algorithms and the SM-UNI algorithm. The grid below illustrates these commonly used mapplets and their OOB replacements. There is a grid comparing values produced by these mapplets and the extended replacements located in the appendix of this document.

Mapplet	Extended Replacement
FIRST_NAME_CASESENSITIVE_V19	dlpx-core:FirstName
LAST_NAME_CASESENSITIVE_V19	dlpx-core:LastName
MIDDLE_NAME_CASESENSITIVE_V19	dlpx-core:FirstName
FULL_NAME_CASESENSITIVE_V19	dlpx-core:FullName
SM_UNI / LONG_STRING	dlpx-core:CM Alpha-Numeric

The extended algorithms produce data in the same format as the mapplets enabling a smooth transition which will not impact most test use cases. Note: Test cases that are dependent on data values (e.g. last name "Smith") may be impacted since the new algorithms will produce different values.

Performance of the new algorithms is much improved over the legacy frameworks. Performance improvements will vary based on algorithm type but all the new frameworks have shown significant performance improvements. Cardinality in the output data for names is also greatly enhanced. The legacy name mapplets and legacy sm_uni mapplet are supported up until the release planned for removal of the legacy framework in the latter half of 2022 providing an 18 month period in which both frameworks are available in the masking engine.

The SM Universal mapplet (known as sm_uni) has been replaced by the new Character Mapping (CM) algorithm. There are three default instances of the CM algorithm: numeric, alpha-numeric and digits. The vast majority of the SM_UNI mapplet distributions will be replaced with **dlpx-core:CM Alpha-Numeric** which will meet most use cases that SM_UNI was implemented for.

Delphix has added new algorithm capability in most releases following the extended framework availability in V6.0.3. The character mapping algorithms, new date algorithms and new name algorithms were introduced in V6.0.5, V6.0.7 and V6.0.8 respectively. The process described in this document will only be required for Delphix masking users that have implemented legacy custom algorithms (mapplets) to meet use cases not met by OOB algorithms during deployment.

Prerequisites

Delphix Version 6.0.8 or greater (V6.0.10 will be required to perform server-wide use of an algorithm directly via API).

- Delphix Masking Toolkit (recommended but not required)
- Inventory assessment to identify mapplet usage (advice included)

Process Summary

- I. Identify custom algorithms and determine replacement algorithms
- II. Backup/export current inventories or create environment copies for applications to be updated
Note: this step is optional
- III. Replace mapplets in inventories
- IV. Verify mapplets are no longer assigned to inventories
- V. Verify domains are updated to new replacement algorithms
- VI. Remove mapplets from masking engine

Process Detail

I. Identify mapplets and determine replacement algorithms

- 1) Determine all legacy custom algorithms (mapplets) that exist on the masking engine via the UI or the API-client or DXM toolkit and inventories they are used in. Record the mapplets in a spreadsheet (example below). The algorithms will have to be removed from inventories prior to removing them from the masking engine. Advice for determining if mapplets exist on the masking engine is found [here](#), below and in the appendix (toolkit).

Identifying mapplets via API-client

1. Navigate to [GET /algorithms/{algorithmName}/usage](#)
2. Enter mapplet name

Parameters		
Parameter	Value	Description
algorithmName	<input type="text" value="FIRST_NAME_MAPPLET"/>	The name of the algorithm

3. Execute API request
4. Capture ruleset IDs to which mapplet is assigned

Response Body

```
{
  "algorithmName": "FIRST_NAME_MAPPLET",
  "columnMetadataIds": [
    19,
    230
  ],
  "fileFieldMetadataIds": [],
  "mainframeDatasetFieldMetadataIds": [],
  "environmentIds": [
    2
  ],
  "rulesetIds": [
    3,
    7
  ],
}
```

- 2) Determine if the mapplets identified in step 1 are used in active masking jobs. If so, note them on your mapping spreadsheet (sample spreadsheet below). The DXM toolkit and/or the masking API-client can be used to determine mapplet assignments in your inventories (advice found in the Appendix). If the mapplets are assigned to inventories they can not be deleted from the server until they are removed from inventories. If the mapplets are **not** assigned to inventories they can be deleted from the server as part of this step and the process is complete.

Note: The DXM toolkit (appendix) will return ruleset names rather than IDs which you may prefer to use to reference the rulesets.

Sample algorithm replacement spreadsheet

Mapplet Name	XML File Name	Ruleset(s) Used In	Replacement Algorithm	Replace Date	Status
FIRST_NAME_CS	First_Name_CaseSensitive_V19.xml	PS_HR_DEV_MSK1	dlpx-core:FirstName	21-Feb-22	Pending
LAST_NAME_CS	Last_Name_CaseSensitive_V19.xml	PS_HR_DEV_MSK1	dlpx-core:LastName	21-Feb-22	Pending
MIDDLE_NAME_CS	Middle_CaseSensitive_V19.xml	PS_HR_DEV_MSK1	dlpx-core:FirstName	21-Feb-22	Pending
FULL_NAME_CS	Full_CaseSensitive_V19.xml	PS_HR_DEV_MSK1	dlpx-core:FullName	21-Feb-22	Pending
LONGSTRING	sm_uni.xml	PS_HR_DEV_MSK1	dlpx-core:CM Alpha-Numeric	21-Feb-22	Pending

- 3) Identify replacement algorithms

The grid above shows the most common mapplets distributed to masking users by Delphix. If the XML file names in the grid are similar to your mapplets you can likely use OOB algorithms above to replace them. This will be true for most Delphix users to which mapplets were distributed.

- a) If your mapplets are not included in the grid above they may be a customization based on these mapplets. Many of the mapplets provided by Delphix were customizations of these mapplets (most notably versions which truncated masked name values for shorter length columns). If your mapplets' XML files have similar names it is likely the replacement algorithms above will handle those requirements since the new name algorithms truncate to column length. The best method to determine suitability of the new OOB algorithms is to run a test with them and evaluate the output.

If there is no suitable OOB algorithm available to replace your mapplets

- b) Contact the mapplet provider to determine how best to replace the mapplet; if the mapplet was provided by Delphix contact your client services representative to schedule a session to review your mapplet use cases. Create a **requirements document** for extended algorithm development similar to sheet below

XML File Name	Use case description	Sample input	Sample output
First_Name_CaseSensitive_V19.xml	Provide masked values in same case as input	John	Ralph
Last_Name_CaseSensitive_V19.xml	Provide masked values in same case as input	Smith	Kramden
Full_Name_CaseSensitive_V19.xml	Provide masked values in same case as input;masked full name = masked first name+masked last name	John Smith	Ralph Kramden
sm_uni.xml	Replace alpha with alpha (preserve input case); replace numeric with numeric; preserve length of string	REX654XY	POB356AB

Note: Extended algorithms are denoted by an '*' next to framework type in Framework column (see below)

Home > Settings > Algorithm

Settings

[Extension support policy](#)

Algorithms Nonconforming Data behavior: Mark job as Succeeded [Learn More](#) [+ Add Algorithm](#)

Algorithms	Name	Framework	Provider	Edit/View/Delete
Custom Algorithms (legacy)	ABC SL	SL*	admin	
Domains	ACCOUNT SL	SL	Built-in	

- c) Map replacement algorithm for all mapplets with OOB replacements in your tracking document

II. Create optional backup of masking inventories employing mapplets: following methods are backup options

- Inventory export for rulesets employing legacy custom algorithms; this is a simple option to record where custom algorithms are assigned prior to replacement; advice found [here](#).
- Environment export via UI or API; this option will create a fully recoverable copy of the environment including the custom algorithms.
- Copy ruleset(s) or environment(s) to another name via UI or API.

note: the ruleset copies will need to be removed from the engine prior to deleting the custom algorithm.

III. Replace legacy custom algorithms in inventories with extended (new framework) algorithms

A. Select method for inventory update

1. API-client - requires V6.0.10, will replace algorithm in all inventories and will update any domains for which the algorithm is the default method. This is the fastest method to change the algorithms; as of V6.0.12 the api supports filtering/replacement by environment and/or ruleset.
2. DXM toolkit - requires DXM V0.9.1; can replace algorithms by environment/ruleset; does not replace algorithm in domain assignment
3. Inventory CSV- can replace algorithms by ruleset
4. Masking UI - can replace individual algorithm assignments in the inventory page
5. Profiler - if you use the Delphix profiler to manage your inventories you can change the default algorithm for your domains and rerun the profiler for rule sets containing mapplets . Note: inventory assignments with ID Method marked as "User" will not be updated by the profiler.

B. Schedule algorithm replacement

- In cases where mapplets are widely used across integrated applications it is recommended to consider the timing of mapplet replacement as values in masked environments will change. Masking execution schedules may vary.
- Assure the replacement window does not overlap with masking execution windows.

C. Execute algorithm replacement

Most replacement activity will employ the masking API-client which is described in this section. (requires V6.0.10 or above). This method will swap out all usage of the algorithm in this engine. For more granular replacement the DXM toolkit or CSV import is recommended. Guidance for use of the DXM Toolkit or CSV import is found in the appendix.

1. Navigate to  [/algorithms/{algorithmName}/usage](#)
2. Using the spreadsheet created in step I enter the algorithm name and replacement algorithm name.

Parameter	Value	Description	Parameter Type	Data Type
algorithmName	LAST_NAME_MAPPLET	The name of the algorithm	path	string
replacementAlgorithmName	dlpx-core:LastName	The name of the replacement algorithm	query	string
ignoreIncompatibleTypes	false (default)	Update usage even when the original and new algorithms have incompatible masking types.	query	boolean
environmentFilter	PS_QA	Only update usage occurring within the specified environment(s). When an environment filter is applied, domain and algorithm reference usage	query	Array[long]

3. Press

4. Execute step 2 and 3 for each mapplet on the replacement list

Note: This process will update the default algorithm for the domain as well as inventories.

IV. Verify inventory changes

- A. Repeat process used in step I to verify that custom algorithms are no longer assigned to inventories

V. Verify Domains are updated to new algorithms

- A. Use your new masking algorithms as default algorithms in your masking domains. This will support new algorithm assignments by the Delphix Profiler.
- B. Domain updates can be made with the following methods:

- UI via Settings / Domain

Edit Domain

Domain Name
LAST_NAME

Algorithm Name
dlpx-core:LastName

Tokenization Algorithm Name
Select Tokenization Algorithm

Cancel Save

- Api-Client

Navigate to [PUT /domains/{domainName}](#) and update default algorithm

Parameter	Value	Description
domainName	LAST_NAME	The name of the domain to update
body	<pre>{ "domainName": "LAST_NAME", "createdBy": "admin", "defaultAlgorithmCode": "dlpx- core:LastName" }</pre>	The updated domain

- DXM Toolkit (see appendix)

VI. Remove Legacy Custom Algorithms from masking engine

- Masking UI- simplest method; navigate to Settings/Custom Algorithms and click the  next to algorithm name
- API-client -Navigate to Algorithms / Delete, enter algorithm name in algorithmName box and execute API request; repeat for all mapplets
- DXM toolkit (see appendix)

Legacy Custom Algorithm (Mapplet) Framework Transition

Over the next several releases of Delphix Masking all of the legacy algorithms will transition to use the new enhanced capabilities. Whereas transition for the legacy custom algorithms (mapplets) require algorithm reassignment in your inventories, the legacy OOB algorithms will not require replacement in most cases. The new algorithms will have the same names as the legacy algorithms they are replacing. Furthermore, values produced by the new algorithm versions will be the same as values produced by the legacy versions. Many of the new algorithms will have additional configurations available but care has been taken to minimize the need to remask data to maintain referential integrity.

In V6.0.12 all existing SL algorithms on the masking engine will automatically transition to the new framework as part of the upgrade process. In addition, customers will be able to migrate their own customized SL algorithms (created with the legacy SL framework after V6.0.12) to the new extended framework via a new algorithm API endpoint. If you have questions on the transition please contact your client services representative.

The projected time frame of transition for the legacy OOB algorithms is found [here](#).

Appendix

DXM toolkit guidance for algorithm replacement

I. Identify mapplets in use on your masking engine(s)

Create a spreadsheet listing all mapplets that exist on the masking engine via the UI or the API-client.

The DXM toolkit can be downloaded [here](#).

```
dxmc algorithms list
```

```
dxmc algorithms list >algo.txt (create file with algorithms in toolkit directory)
```

- The commands above will produce a list of all algorithms on the masking engine
- Mapplets will appear in output of above command as algorithm type "CUSTOM ALGORITHM"

Ex. Get list of all LAST_NAME_CS inventory assignments

```
dxmc column list --alname LAST_NAME_CS --engine ME608
```

or

```
dxmc column list --alname LAST_NAME_CS --engine ME608 >last_name.txt
```

Engine name	Environment name	Ruleset name	Metadata name	Column name
Data type	Domain name	Alg name		
=====	=====	=====	=====	=====
ME608	Oracle	Oracle_RS1	EMPLOYEE	LAST NAME
VARCHAR2(50)	LAST NAME	LAST_NAME_CS		

Note: File rulesets will display with a file format name rather than a table name; a bug was reported for current version of DXM (0.9.1) which impacts updating algorithm for file rulesets with DXM. For file inventories the mapplet has to be replaced using the API Client or one of the other methods described above.

II. Create optional backup of masking inventories employing mapplets (see Step II in main document for options)

III. Replace algorithms using DXM toolkit

DXM toolkit Command line examples

1. To replace all uses of the mapplet on the server

```
dxmc column replace --alname LAST_NAME_MAPPLET --newalname dlpx-core:LastName  
--newdomain LAST_NAME
```

2. To replace all uses of the mapplet in a specific environment

```
dxmc column replace --alname LAST_NAME_MAPPLET --newalname dlpX-core:LastName  
--newdomain LAST_NAME --envname ORACLE_DEV
```

3. To replace the mapplet in a specific ruleset

```
dxmc column replace --alname LAST_NAME_MAPPLET --newalname dlpX-core:LastName  
--newdomain LAST_NAME --rulesetname CUST_RS1
```

4. To replace the custom algorithms as the default algorithm for a domain

```
dxmc domain update --domainname LAST_NAME --alname dlpX-core:LastName
```

IV. Verify inventory changes

- Repeat Step I in this section to assure the algorithm is no longer assigned in rulesets

V. Verify Domains are updated to new default algorithms

- Verify default algorithms are updated via UI or API-client (see Step V in main document above)

VII. Remove custom algorithms from masking engine

- DXM toolkit does not currently support deletion of algorithms (please refer to Step VI in main document for options)

Mapplet/Extended algorithm masked value comparison

Unmasked Input

	FIRST_NAME	LAST_NAME	MIDDLE_NAME	FULL_NAME	NAME_DISPLAY
1	robert	patten	foghorn	Patten, Robert F	Robert Patten
2	Linh	Bui	Tin	Bui, Linh Tin	Mr Linh Tin Bui
3	JOHN	SMITH	CONNER	Smith, John	JOHN CONNER SMITH
4	John	Conner	KANSAS	Conner, John KANSAS	Dr. John Conner
5	Larry	Bird	JOE	Bird, Larry Joe	Larry Joe Bird
6	Aidan	Aguilar	Mel	Aguilar, Aidan	Aidan Mel Aguilar
7	AIDAN	AGUILAR	MEL	AGUILAR, AIDAN	AIDAN M. AGUILAR
8	larry	bird	joe	bird, larry joe	larry j. bird jr
9	LARRY	BIRD	JOE	BIRD, LARRY JOE	LARRY J. BIRD
10	Jesse	Jackson	Jerry	Unknow	Unknow

Name Mapplets V19 masked output

	FIRST_NAME	LAST_NAME	MIDDLE_NAME	FULL_NAME	NAME_DISPLAY
1	leonardus	vantrease	erwan	Vantrease, Leonardus A	Leonardus Vantrease
2	Demetra	Asato	Jacki	Asato, Demetra Jacki	Demetra Jacki Asato
3	CONNOR	BECCARIA	STEVE	Beccaria, Connor	CONNOR STEVE BECCARIA
4	Connor	Shorey	EZELL	Shorey, Connor EZELL	Connor Shorey
5	Alanzo	Stady	ELISSA	Stady, Alanzo Elissa	Alanzo Elissa Stady
6	Yousra	Hysom	Buster	Hysom, Yousra	Yousra Buster Hysom
7	YOUSRA	HYSOM	BUSTER	HYSOM, YOUSRA	YOUSRA I. HYSOM
8	alanzo	stady	elissa	stady, alanzo elissa	alanzo c. stady
9	ALANZO	STADY	ELISSA	STADY, ALANZO ELISSA	ALANZO C. STADY
10	Jaye	Martiez	Rolande	Pathak	Pathak

New OOB name algorithm output

	FIRST_NAME	LAST_NAME	MIDDLE_NAME	FULL_NAME	NAME_DISPLAY
	earline	audley	samantha	Audley, Pearlina C	Pearline Audley
	athlene	Richmond	Glenda	Richmond, Kathlene Glenda	Zula Kathlene Glenda Richmond
	JUCIANO	CAMPBELL	ELZA	Campbell, Luciano	LUCIANO ELZA CAMPBELL
	uciano	Atherton	KYOKO	Atherton, Luciano Kyoko	Nenita Luciano Atherton
	ewel	Reynold	SOON	Reynold, Jewel Soon	Jewel Soon Reynold
	ennie	Rowley	Demetrius	Rowley, Gennie	Gennie Demetrius Rowley
	ENNIE	ROWLEY	DEMETRIUS	ROWLEY, GENNIE	GENNIE M ROWLEY
	ewel	reynold	soon	reynold, jewel soon	jewel a oretha borland
	EWEL	REYNOLD	SOON	REYNOLD, JEWEL SOON	JEWEL A REYNOLD
	malia	Reese	Darnell	Janay	Janay

SM_UNI and CM Alpha-numeric

	UNMASKED	SM_UNI	CM_ALPHANUMERIC
1	213612345	284035176	776533758
2	L45699-9999	R59073-4509	W32846-9914
3	ABC! ?050004	GGG! ?776454	PKO! ?230485
4	John Smith	Pnho Iqnzo	Ghye Oiabc
5	X-Y-Z*321	D-C-B*190	Q-N-S*503
6	123123123	174749492	583752394
7	ABCabcXYZxyz---GGG	GGGbdFXXXegi---XWV	BDYbubSJVsir---KXT
8	?^~Breakfast is Tasty	?^~Etheplhac te Jpikq	?^~Ogkvbphwg mh Zfvhe
9	67 89 44 55 676788 99 4321	63 37 27 45 549050 09 0546	79 16 08 63 159146 67 2962