DELPHIX

Masking Algorithm Guide

August 2018

Masking Algorithm Guide
You can find the most up-to-date technical documentation at:
docs.delphix.com The Delphix Web site also provides the latest product updates.
If you have comments about this documentation, submit your feedback to: infodev@delphix.com

Delphix Corp.
1400 Seaport Blvd, Suite 200
Redwood City, CA 94063

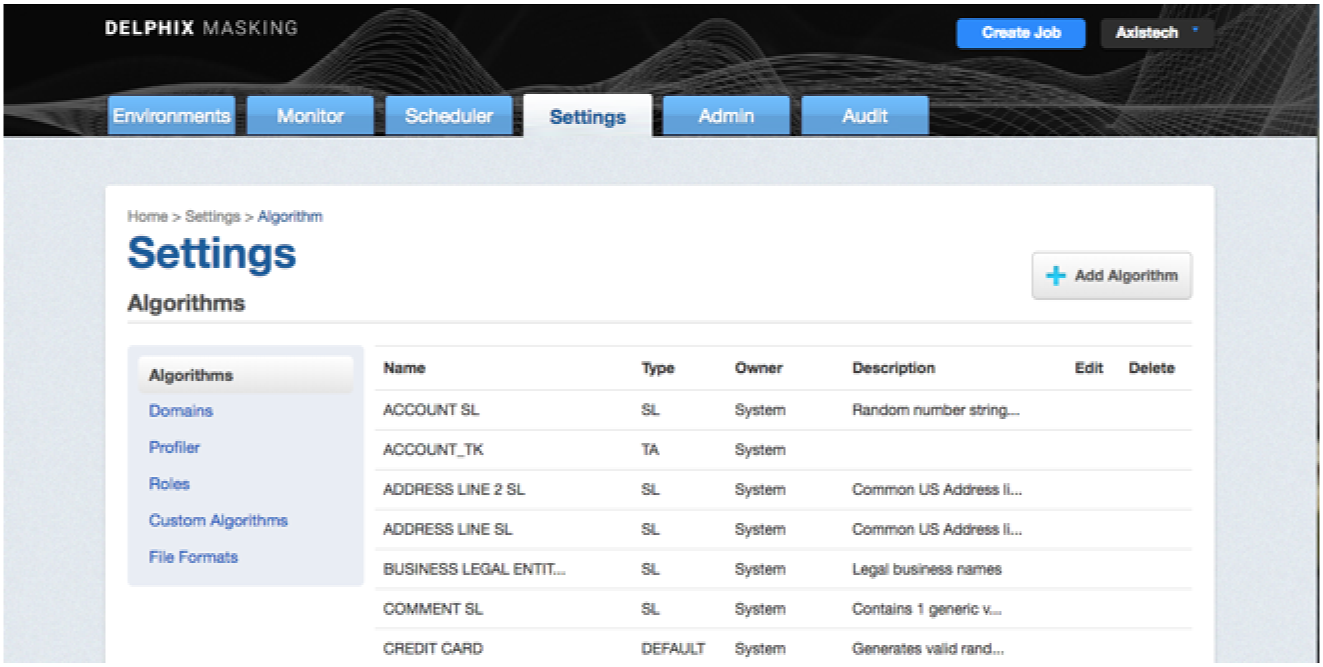
Managing Algorithm Settings

An integral part of the data masking process is to use algorithms to mask each data element. You specify which algorithm to use on each individual data element (domain) on the Masking's tab. There, you define a unique domain for each element and then associate the classification and algorithm you want to use for each domain. Use the Algorithm settings tab to create or delete algorithms.

Algorithm Settings Tab

The **Algorithm** tab displays algorithm Names along with Type and Description. This is where you add (or create) new algorithms. The default Delphix Masking Engine algorithms and any algorithms you have defined appear on this tab.

- All algorithm values are stored encrypted. These values are only decrypted during the masking process.



Name	Type	Owner	Description	Edit	Delete
ACCOUNT SL	SL	System	Random number string...		
ACCOUNT_TK	TA	System			
ADDRESS LINE 2 SL	SL	System	Common US Address li...		
ADDRESS LINE SL	SL	System	Common US Address li...		
BUSINESS LEGAL ENTIT...	SL	System	Legal business names		
COMMENT SL	SL	System	Contains 1 generic v...		
CREDIT CARD	DEFAULT	System	Generates valid rand...		

Algorithm Settings Tab

Adding New Delphix Masking Engine Algorithms

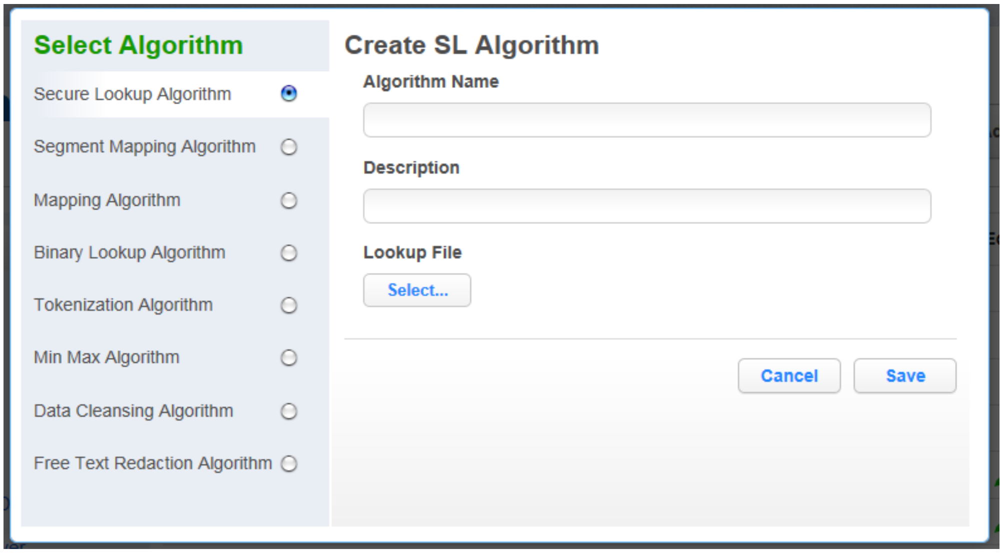
If none of the default Delphix Masking Engine algorithms meet your needs, you might want to create a new algorithm.

The Delphix Masking Engine Algorithm Frameworks give you the ability to quickly and easily define the algorithms you want, directly on the Settings page. Then, you can immediately propagate them. Anyone in your organization who has the Delphix Masking Engine can then access the information.

> Administrators can update **system**-defined algorithms. User-defined algorithms can be accessed by all users and updated by the owner/user who created the algorithm.

To add an algorithm:

1. In the upper right-hand corner of the **Algorithm** settings tab, click **Add Algorithm**.



2. Select an algorithm type.

3. Complete the form to the right to name and describe your new algorithm.

4. Click **Save.**

Choosing an Algorithm Type

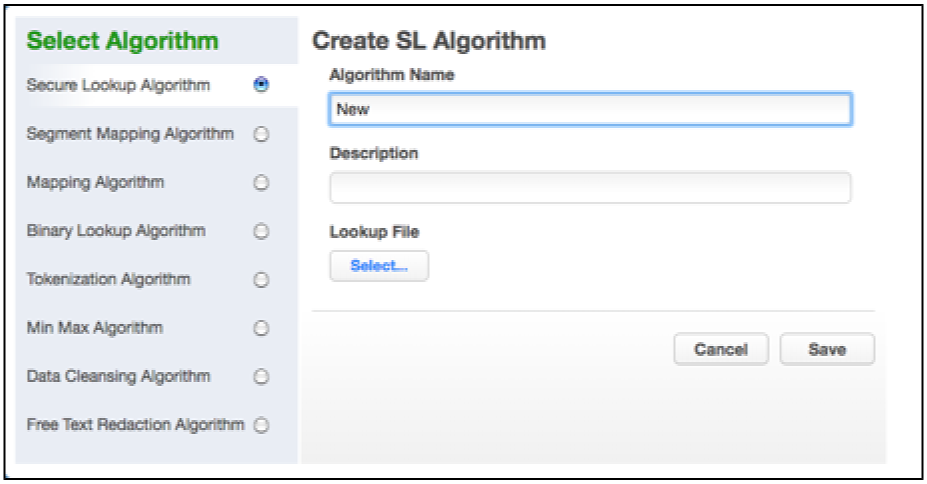
The Delphix Masking Engine offers 35 individual algorithms from which to choose, so you can mask data according to your specific needs. Each algorithm is built using one of eight frameworks, or algorithm types. The descriptions below will help you select which algorithm type is appropriate for the way that you want to mask data. They appear in order of their popularity.

Secure Lookup Algorithm

Secure lookup is the most commonly used type of algorithm. It is easy to generate and works with different languages. When this algorithm replaces real, sensitive data with fictional data, it is possible that it will create repeating data patterns, known as “collisions.” For example, the names “Tom” and “Peter” could both be masked as “Matt.” Because names and addresses naturally recur in real data, this mimics an actual data set. However, if you want the masking engine to mask all data into unique outputs, you should use segment mapping.

To add a secure lookup algorithm:

1. In the upper right-hand corner of the **Algorithm** tab, click **Add Algorithm**.
2. Choose **Secure Lookup Algorithm**. The Create SL Algorithm pane appears.


Select Algorithm
Secure Lookup Algorithm ●
Segment Mapping Algorithm ○
Mapping Algorithm ○
Binary Lookup Algorithm ○
Tokenization Algorithm ○
Min Max Algorithm ○
Data Cleansing Algorithm ○
Free Text Redaction Algorithm ○
Create SL Algorithm
Lookup File
Select...
Cancel
Save

Algorithm Name: New

Description:

3. Enter a **Algorithm Name**.

> This name MUST be unique.

4. Enter a **Description**.
5. Specify a **Lookup File**.
   This file is a single list of values. It does not require a header. Make sure there are no spaces or returns at the end of the last line in the file. The following is sample file content:

Example Lookup File

```
Smallville
Clarkville
Farmville
Townville
Cityname
Citytown
Towneaster
```

6. When you are finished, click **Save**.
7. Before you can use the algorithm (specify it in a profiling or masking job), you must add it to a domain.

> **Note**
> The masking engine supports lookup files saved in ASCII or UTF-8 format only. If the lookup file contains foreign alphabet characters, the file must be saved in UTF-8 format with no BOM (Byte Order Marker) for Masking Engine to read the Unicode text correctly. Some applications, e.g. Notepad on Windows, write a BOM (Byte Order Marker) at the beginning of Unicode files which irritates the masking engine and will lead to SQL update or insert errors when trying to run a masking job that applies a Secure Lookup algorithm that has been created based on a UTF-8 file that included a BOM.

Segment Mapping Algorithm

Segment mapping algorithms produce no overlaps or repetitions in the masked data. They let you create unique masked values by dividing a target value into separate segments and masking each segment individually.

You can mask up to a maximum of 36 values using segment mapping. You might use this method if you need columns with unique values, such as Social Security Numbers, primary key columns, or foreign key columns. When using segment mapping algorithms for primary and foreign keys, in order to make sure they match, you must use the same segment mapping algorithm for each. You can set the algorithm to produce alphanumeric results (letters and numbers) or only numbers.

With segment mapping, you can set the algorithm to ignore specific characters. For example, you can choose to ignore dashes [-] so that the same Social Security Number will be identified no matter how it is formatted. You can also preserve certain values. For example, to increase the randomness of masked values, you can preserve a single number such as 5 wherever it occurs. Or if you want to leave some information unmasked, such as the last four digits of Social Security numbers, you can preserve that information.

Segment Mapping Example

Perhaps you have an account number for which you need to create a segment mapping algorithm. You can separate the account number into segments, preserving the first two-character segment, replacing a segment with a specific value, and preserving a hyphen. The following is a sample value for this account number:

NM831026-04

Where:

- **NM** is a plan code number that you want to preserve, always a two-character alphanumeric code.
- **831026** is the uniquely identifiable account number. To ensure that you do not inadvertently create actual account numbers, you can replace the first two digits with a sequence that never appears in your account numbers in that location. (For example, you can replace the first two digits with 98 because 98 is never used as the first two digits of an account number.) To do that, you want to split these six digits into two segments.
- **-04** is a location code. You want to preserve the hyphen and you can replace the two digits with a number within a range (in this case, a range of 1 to 77).

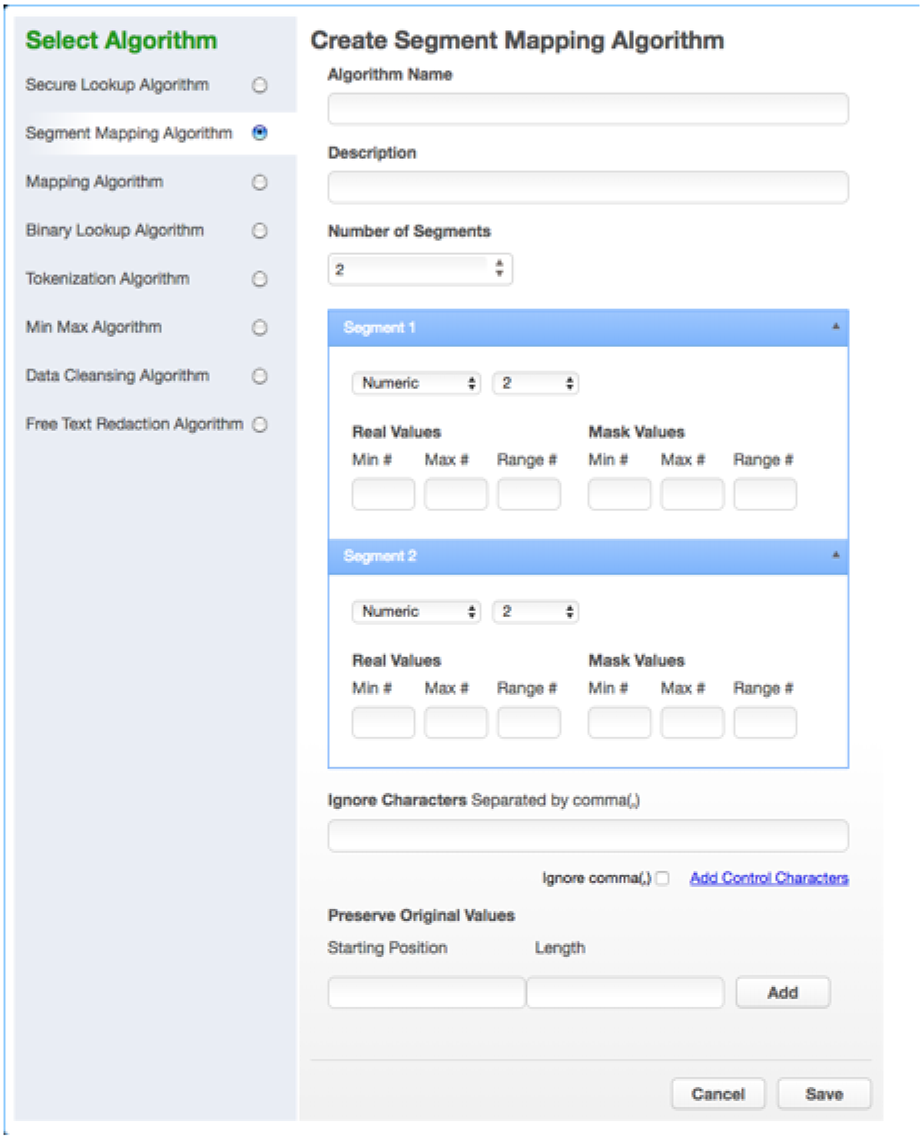
Procedure for Defining Segments

1. Choose **3** for **No. of Segment**. Remember, you do NOT count the segment(s) you want to preserve.
2. Preserve the first two characters ("NM" in the sample value). Under **Preserve Original Values**:
   a. For **Starting position**, enter **1**.
   b. For **Length**, enter **2**.
3. Define the next two-digit segment ("83" in sample value) to always be 98 or 99.
   a. For **Segment 1**, select **Type** > **Numeric**.
   b. For **Length**, select **2**.
   c. For **Mask Values Range#**, specify 98,99.
4. Define the next four-digit segment ("1026" in sample value).
   a. For **Segment 2**, select **Type** > **Numeric**.
   b. For **Length**, select **4**.
   c. Leave range fields empty.
   d. Click **Add** to the right of **Preserve Original Values**.
5. Preserve the hyphen.
   a. For **Starting position**, enter **9**.
   b. For **Length**, enter **1**.
6. Define the last two-digit segment ("04" in sample value).
   a. For **Segment 3**, select **Type** > **Numeric**.
   b. For **Length**, select **2**.
   c. For **Mask Values Min#**, enter **1**.
   d. For **Mask Values Max#**, enter **77**.

The sample value NM831026-04 might be masked to NM981291-77.

Segment Mapping Procedure

1. In the upper right-hand region of the **Algorithm** tab, click **Add Algorithm**.
2. Select **Segment Mapping Algorithm**. The Create Segment Mapping Algorithm pane appears.



3. Enter an **Algorithm Name** as shown in the Create pane.
4. Enter a **Description.**
5. From the **No. of Segment** drop-down menu, select how many segments you want to mask.

   This number does NOT include the values you want to preserve.

   The minimum number of segments is 2; the maximum is 9.
   A box appears for each segment.
6. For each segment, choose the **Type** of segment from the dropdown: **Numeric** or **Alphanumeric**.

   Numeric segments are masked as whole segments. **Alphanumeric** segments are masked by individual character.

7. For each segment, select its **Length** (number of characters) from the drop-down menu. The maximum is 4.
8. Optionally, for each segment, specify range values. You might need to specify range values to satisfy particular application requirements, for example. See details below.
9. **Preserve Original Values** by entering **Starting position** and **length** values. (Position starts at 1.) For example, to preserve the second, third, and fourth values, enter Starting position **2** and length **3**.
   a. If you need additional value fields, click **Add**.
10. When you are finished, click **Save**.
11. Before you can use the algorithm (specify it in a profiling or masking job), you must add it to a domain. If you are not using the Masking Engine Profiler to create your inventory, you do not need to associate the algorithm with a domain.

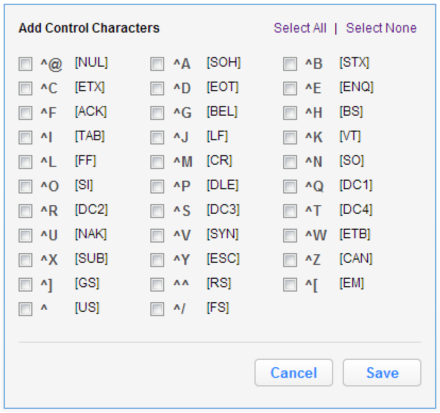
Specifying Range Values

You can specify ranges for **Real Values** and **Mask Values**. With Real Values ranges, you can specify all the possible real values to map to the ranges of masked values. Any values NOT listed in the Real Values ranges would then mask to themselves.

Specifying range values is optional. If you need unique values (for example, masking a unique key column), you MUST leave the range values blank. If you plan to certify your data, you must specify range values.

When determining a numeric or alphanumeric range, remember that a narrow range will likely generate duplicate values, which will cause your job to fail.

1. To ignore specific characters, enter one or more characters in the **Ignore Character List** box. Separate values with a comma.
2. To ignore the comma character (,), select the **Ignore comma (,)** check box.
3. To ignore control characters, select **Add Control Characters**.
   The **Add Control Characters** window appears.



☐	^@	[NUL]	☐	^A	[SOH]	☐	^B	[STX]
☐	^C	[ETX]	☐	^D	[EOT]	☐	^E	[ENQ]
☐	^F	[ACK]	☐	^G	[BEL]	☐	^H	[BS]
☐	^I	[TAB]	☐	^J	[LF]	☐	^K	[VT]
☐	^L	[FF]	☐	^M	[CR]	☐	^N	[SO]
☐	^O	[SI]	☐	^P	[DLE]	☐	^Q	[DC1]
☐	^R	[DC2]	☐	^S	[DC3]	☐	^T	[DC4]
☐	^U	[NAK]	☐	^V	[SYN]	☐	^W	[ETB]
☐	^X	[SUB]	☐	^Y	[ESC]	☐	^Z	[CAN]
☐	^]	[GS]	☐	^^	[RS]	☐	^[	[EM]
☐	^	[US]	☐	^/	[FS]			

4. Select the individual control characters that you would like to ignore, or choose **Select All** or **Select None**.
5. When you are finished, click **Save**.
   You are returned to the Segment Mapping pane.

Numeric segment type

- **Min#** — A number; the first value in the range. Value can be 1 digit or up to the length of the segment. For example, for a 3-digit segment, you can specify 1, 2, or 3 digits. Acceptable characters: 0-9.
- **Max#** — A number; the last value in the range. Value should be the same length as the segment. For example, for a 3-digit segment, you should specify 3 digits. Acceptable characters: 0-9.
- **Range#** — A range of numbers; separate values in this field with a comma (,). Value should be the same length as the segment. For example, for a 3-digit segment, you should specify 3 digits. Acceptable characters: 0-9.

> If you do not specify a range, the Masking Engine uses the full range. For example, for a 4-digit segment, the Masking Engine uses 0-9999.

Alphanumeric segment type

- **Min#** — A number from 0 to 9; the first value in the range.
- **Max#** — A number from 0 to 9; the last value in the range.
- **MinChar** — A letter from A to Z; the first value in the range.
- **MaxChar** — A letter from A to Z; the last value in the range.
- **Range#** — A range of alphanumeric characters; separate values in this field with a comma (,). Individual values can be a number from 0 to 9 or an uppercase letter from A to Z. (For example, B,C,J,K,Y,Z or AB,DE.)

> If you do not specify a range, the Masking Engine uses the full range (A-Z, 0-9). If you do not know the format of the input, leave the range fields empty. If you know the format of the input (for example, always alphanumeric followed by numeric), you can enter range values such as A2 and S9.

Mapping Algorithm

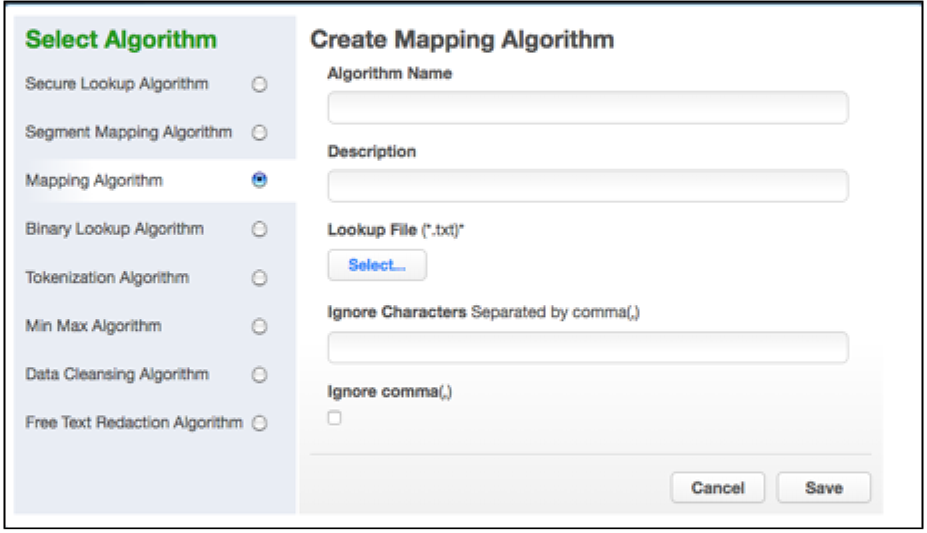
A mapping algorithm allows you to state what values will replace the original data. It sequentially maps original data values to masked values that are pre-populated to a lookup table through the Masking Engine user interface. There will be no collisions in the masked data, because it always matches the same input to the same output. For example “David” will always become “Ragu,” and “Melissa” will always become “Jasmine.” The algorithm checks whether an input has already been mapped; if so, the algorithm changes the data to its designated output.

You can use a mapping algorithm on any set of values, of any length, but you must know how many values you plan to mask. You must supply AT MINIMUM the same number of values as the number of unique values you are masking; more is acceptable. For example, if there are 10,000 unique values in the column you are masking you must give the mapping algorithm AT LEAST 10,000 values.

> When you use a mapping algorithm, you cannot mask more than one table at a time. You must mask tables serially.

To add a mapping algorithm:

1. In the upper right-hand corner of the **Algorithm** tab, click **Add Algorithm**.
2. Select **Mapping Algorithm**.
   The **Create Mapping Algorithm** pane appears.



3. Enter a **Rule Name**. This name MUST be unique.
4. Enter a **Description**.
5. Specify a **Lookup File (**.txt){*}.
   The value file must have NO header. Make sure there are no spaces or returns at the end of the last line in the file. The following is sample file content. Notice that there is no header and only a list of values.

```
Smallville
Clarkville
Farmville
Townville
Cityname
Citytown
Towneaster
```

6. To ignore specific characters, enter one or more characters in the **Ignore Character List** box. Separate values with a comma.
7. To ignore the comma character (,), select the **Ignore comma (,)** check box.
8. When you are finished, click **Save**.

Before you can use the algorithm by specifying it in a profiling or masking job, you must add it to a domain. If you are not using the Masking Engine Profiler to create your inventory, you do not need to associate the algorithm with a domain.

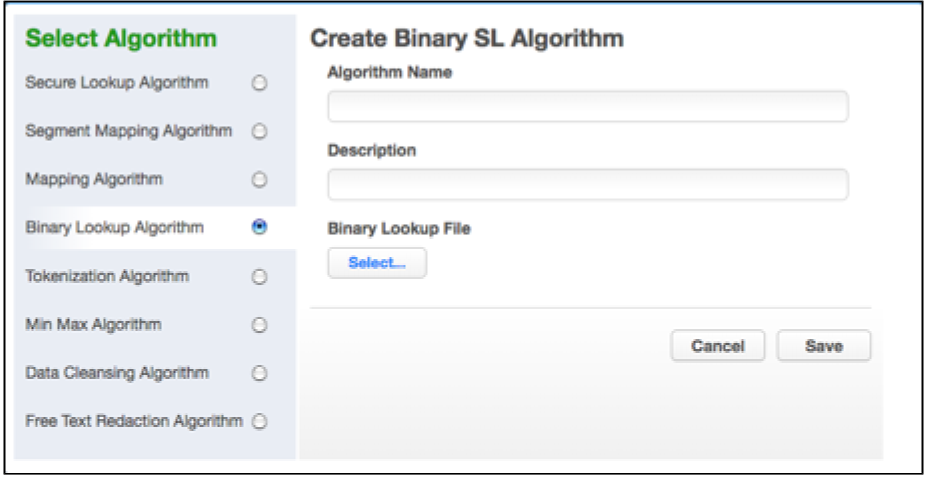
See Adding New Domains.

Binary Lookup Algorithm

A Binary Lookup Algorithm is much like the Secure Lookup Algorithm, but is used when entire files are stored in a specific column. This algorithm replaces objects that appear in object columns. For example, if a bank has an object column that stores images of checks, you can use a binary lookup algorithm to mask those images. The Delphix Engine cannot change data within images themselves, such as the names on X-rays or driver's licenses. However, you can replace all such images with a new, fictional image. This fictional image is provided by the owner of the original data.

To add a binary lookup algorithm:

1. At the top right of the **Algorithm** tab, click **Add Algorithm**.
2. Select **Binary Lookup Algorithm**.
   The Binary SL Rule pane appears.

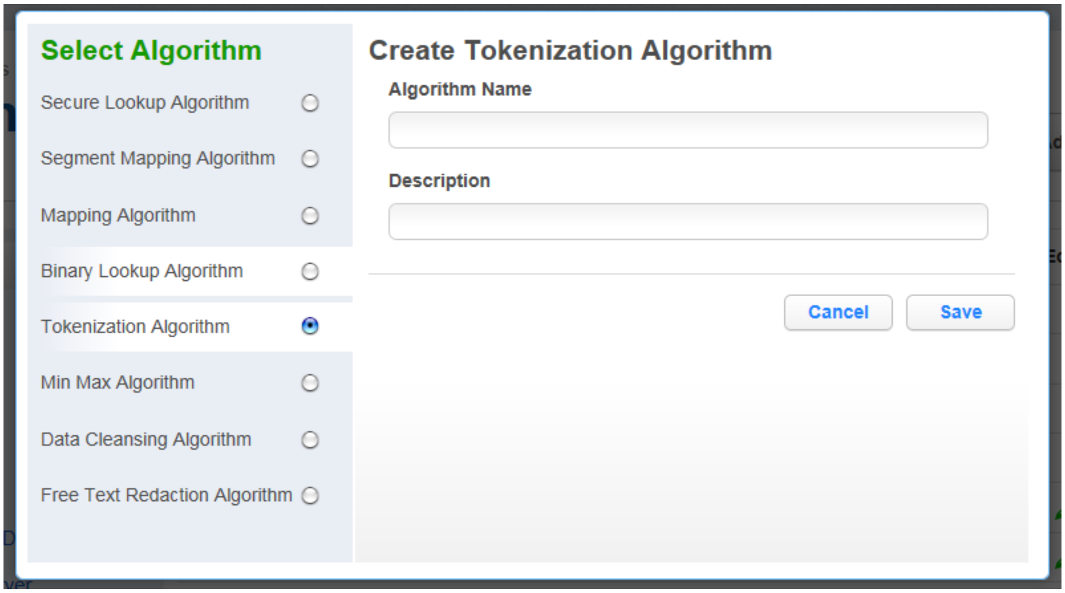

Select Algorithm
Secure Lookup Algorithm ○
Segment Mapping Algorithm ○
Mapping Algorithm ○
Binary Lookup Algorithm ●
Tokenization Algorithm ○
Min Max Algorithm ○
Data Cleansing Algorithm ○
Free Text Redaction Algorithm ○
Create Binary SL Algorithm
Algorithm Name
Description
Binary Lookup File
Select...
Cancel
Save

3. Enter a **Rule Name**.
4. Enter a **Description**.
5. Select a **Binary Lookup File** on your filesystem.
6. Click **Save**.

Tokenization Algorithm

A tokenization algorithm is the only type of algorithm that allows you to reverse its masking. For example, you can use a tokenization algorithm to mask data before you send it to an external vendor for analysis. The vendor can then identify accounts that need attention without having any access to the original, sensitive data. Once you have the vendor's feedback, you can reverse the masking and take action on the appropriate accounts.

Like mapping, a tokenization algorithm creates a unique token for each input such as "David" or "Melissa." The actual data (for example, names and addresses) are converted into tokens that have similar properties to the original data – such as text and length – but no longer convey any meaning. The Delphix Masking Engine stores both the token and the original so that you can reverse masking later.

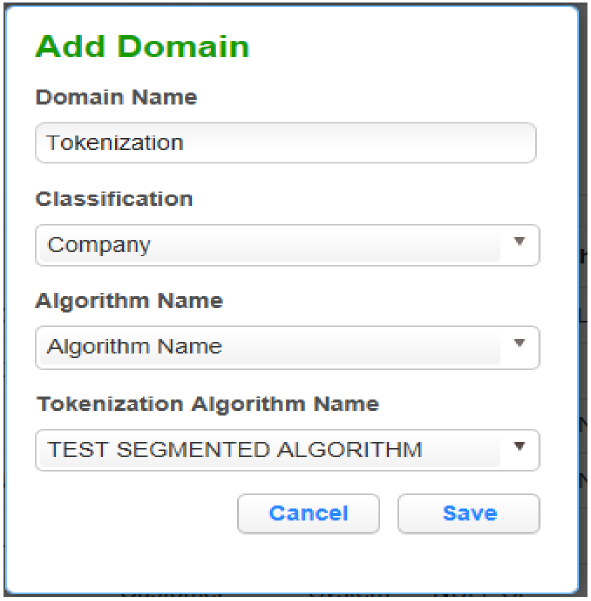

Select Algorithm
Secure Lookup Algorithm ○
Segment Mapping Algorithm ○
Mapping Algorithm ○
Binary Lookup Algorithm ○
Tokenization Algorithm ●
Min Max Algorithm ○
Data Cleansing Algorithm ○
Free Text Redaction Algorithm ○
Create Tokenization Algorithm
Algorithm Name
Description
Cancel
Save

To add a Tokenization algorithm:

1. Enter algorithm **Name**.
2. Enter a **Description**.
3. Click **Save**.

Once you have created an algorithm, you will need to associate it with a domain.

1. Navigate to the **Home>Settings>Domains** page and click **Add Domain**. You will see the popup below:



Add Domain

Domain Name: Tokenization

Classification: Company

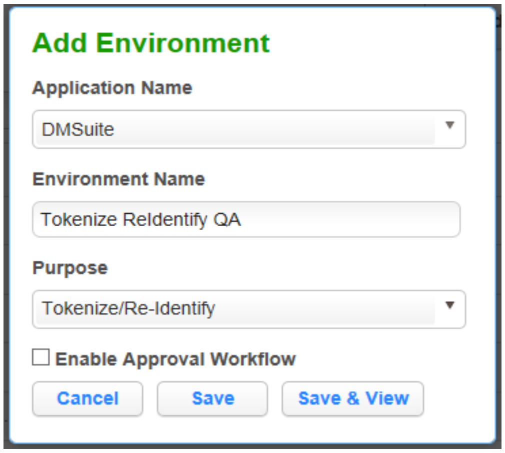
Algorithm Name: Algorithm Name

Tokenization Algorithm Name: TEST SEGMENTED ALGORITHM

2. Enter a domain name.
3. From the **Tokenization Algorithm Name** drop-down menu, select your algorithm.

Create a Tokenization Environment

1. On the home page, click **Environments**.
2. Click **Add Environment**.



Add Environment

Application Name: DMSuite

Environment Name: Tokenize ReIdentify QA

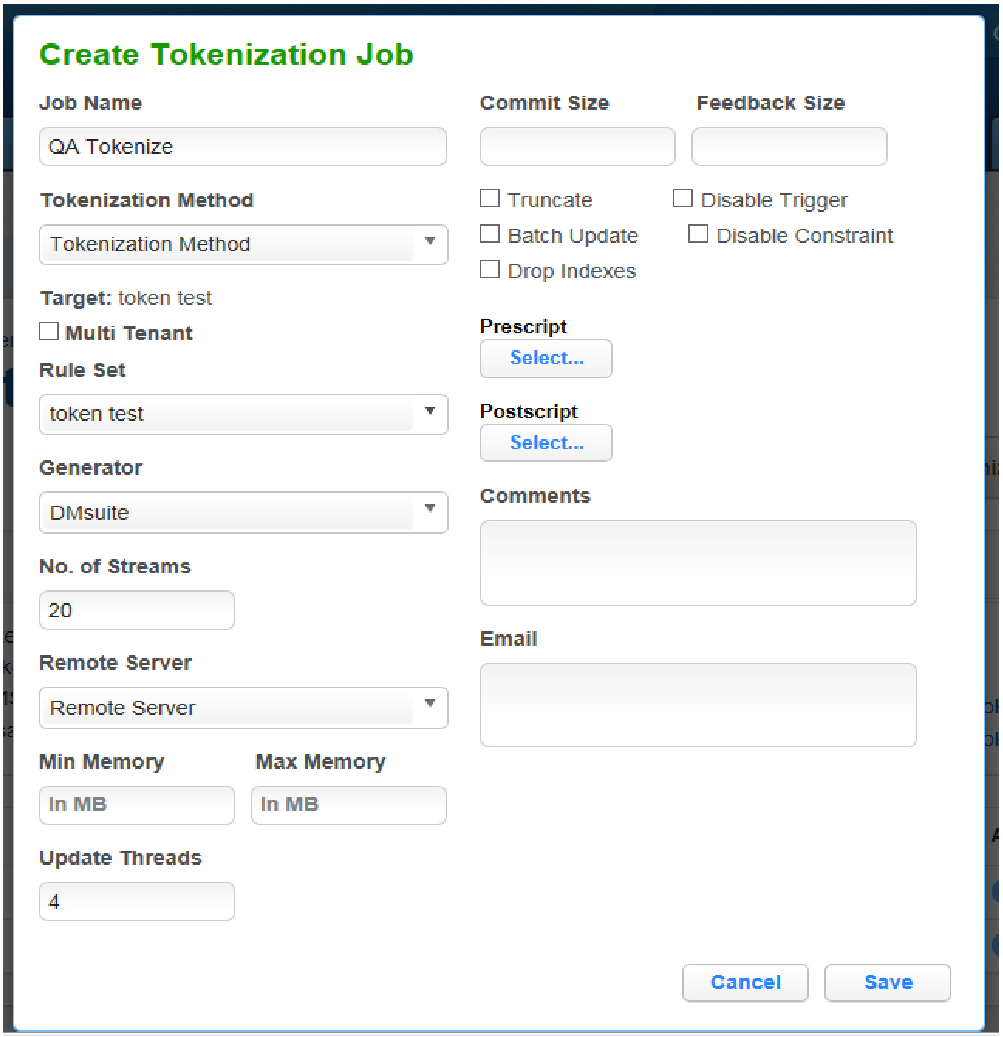
Purpose: Tokenize/Re-Identify

☐ Enable Approval Workflow

3. For **Purpose**, select **Tokenize/Re-Identify**.
4. Click **Save**.

> This environment will be used to re-identify your data when required.

5. Set up a Tokenize job using tokenization method. Execute the job.



Create Tokenization Job

Job Name: QA Tokenize

Commit Size:

Feedback Size:

Tokenization Method: Tokenization Method

□ Truncate
□ Disable Trigger
□ Batch Update
□ Disable Constraint
□ Drop Indexes

Target: token test

□ Multi Tenant

Rule Set: token test

Prescript: Select...

Postscript: Select...

Generator: DMsuite

Comments:

No. of Streams: 20

Email:

Remote Server: Remote Server

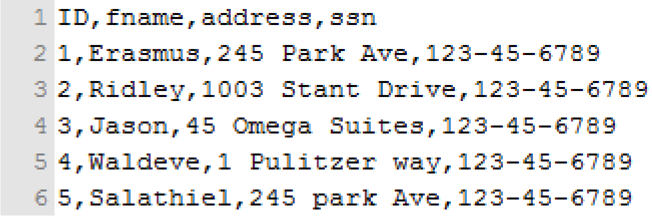
Min Memory: In MB

Max Memory: In MB

Update Threads: 4

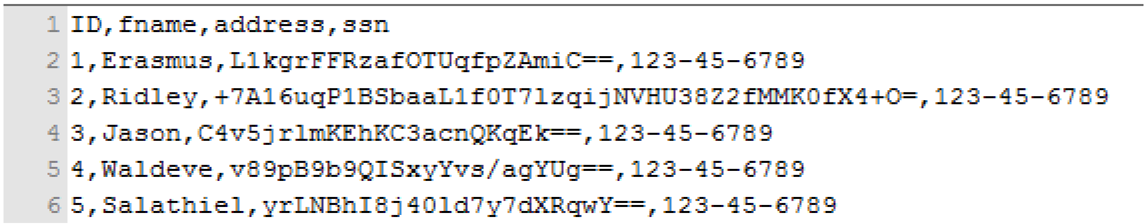
Here is a snapshot of the data before and after Tokenization to give you an idea of what the it will look like.

Before Tokenization



```
ID,fname,address,ssn
1,Erasmus,245 Park Ave,123-45-6789
2,Ridley,1003 Stant Drive,123-45-6789
3,Jason,45 Omega Suites,123-45-6789
4,Waldeve,1 Pulitzer way,123-45-6789
5,Salathiel,245 park Ave,123-45-6789
```

After Tokenization

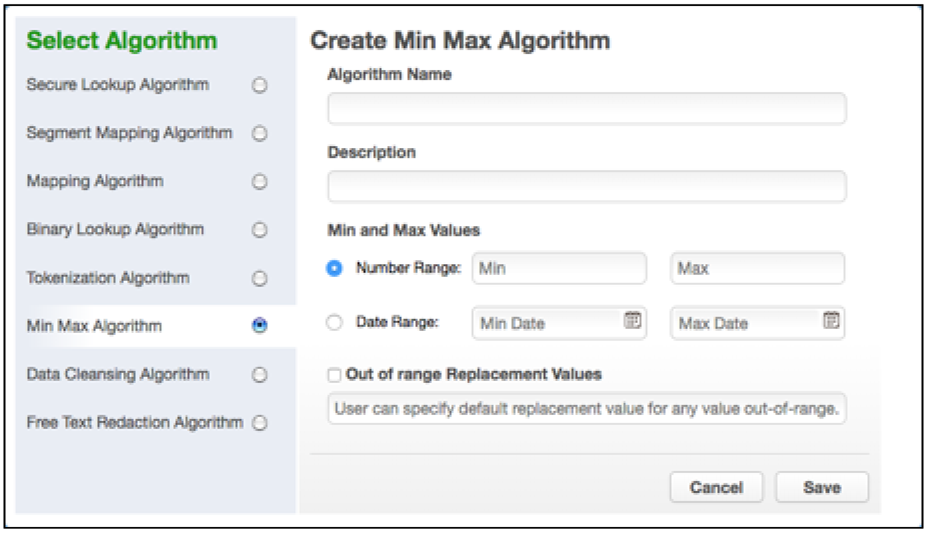


```
ID,fname,address,ssn
1,Erasmus,L1kgrFFRzafOTUqfpZAmiC==,123-45-6789
2,Ridley,+7A16uqP1BSbaaL1f0T7lzqijNVHU38Z2fMMK0fX4+O=,123-45-6789
3,Jason,C4v5jrlmKEhKC3acnQKqEk==,123-45-6789
4,Waldeve,v89pB9b9QISxyYvs/agYUg==,123-45-6789
5,Salathiel,yrLNBhI8j40ld7y7dXRqwY==,123-45-6789
```

Min Max Algorithm

The Delphix Masking Engine provides a "Min Max Algorithm" to normalize data within a range – for example, 10 to 400. Values that are extremely high or low in certain categories allow viewers to infer someone's identity, even if their name has been masked. For example, a salary of $1 suggests a company's CEO, and some age ranges suggest higher insurance risk. You can use a min max algorithm to move all values of this kind into the midrange. This algorithm allows you to make sure that all the values in the database are within a specified range.

If the **Out of range Replacement Values** checkbox is selected, a default value is used when the input cannot be evaluated.

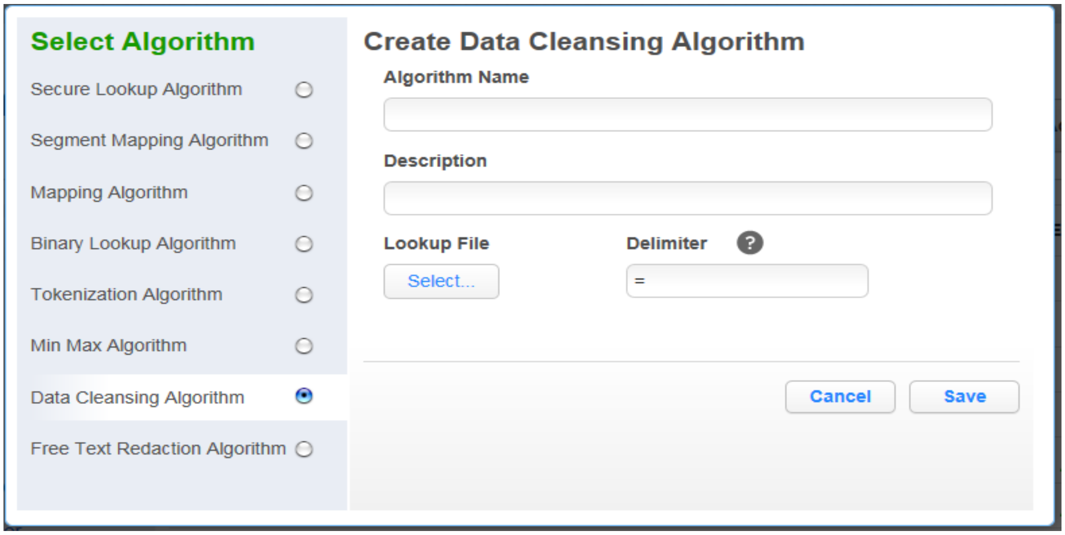

Select Algorithm
Secure Lookup Algorithm ○
Segment Mapping Algorithm ○
Mapping Algorithm ○
Binary Lookup Algorithm ○
Tokenization Algorithm ○
Min Max Algorithm ●
Data Cleansing Algorithm ○
Free Text Redaction Algorithm ○
Create Min Max Algorithm
Algorithm Name
Description
Min and Max Values
● Number Range: Min Max
○ Date Range: Min Date Max Date
□ Out of range Replacement Values
User can specify default replacement value for any value out-of-range.
Cancel
Save

1. Enter the **Algorithm Name**.
2. Enter a **Description**.
3. Enter **Min Value** and **Max Value**.
4. Click **Out of range Replacement Values**.
5. Click **Save**.

Example: Age less than 18 years - enter Min Value 0 and Max Value 18

Data Cleansing Algorithm

A data cleansing algorithm does not perform any masking. Instead, it standardizes varied spellings, misspellings, and abbreviations for the same name. For example, "Ariz," "Az," and "Arizona" can all be cleansed to "AZ." Use this algorithm if the target data needs to be in a standard format prior to masking.


Select Algorithm
Secure Lookup Algorithm ○
Segment Mapping Algorithm ○
Mapping Algorithm ○
Binary Lookup Algorithm ○
Tokenization Algorithm ○
Min Max Algorithm ○
Data Cleansing Algorithm ●
Free Text Redaction Algorithm ○
Create Data Cleansing Algorithm
Algorithm Name
Description
Lookup File
Select...
Delimiter ?
=
Cancel
Save

1. Enter Algorithm **Name**.
2. Enter a **Description**.
3. Select **Lookup File** location.
4. Enter default **Delimiter**. Key and Value separator is =. You can change this to match the lookup file.
5. Click **Save**.

Below is an example of a lookup input file. It does not require a header. Make sure there are no spaces or returns at the end of the last line in the file. The following is sample file content:

Example Lookup File

```
NYC=NY
NY City=NY
New York=NY
Manhattan=NY
```

Free Text Algorithm

A free text redaction algorithm helps you remove sensitive data that appears in free-text columns such as “Notes.” This type of algorithm requires some expertise to use, because you must set it to recognize sensitive data within a block of text.

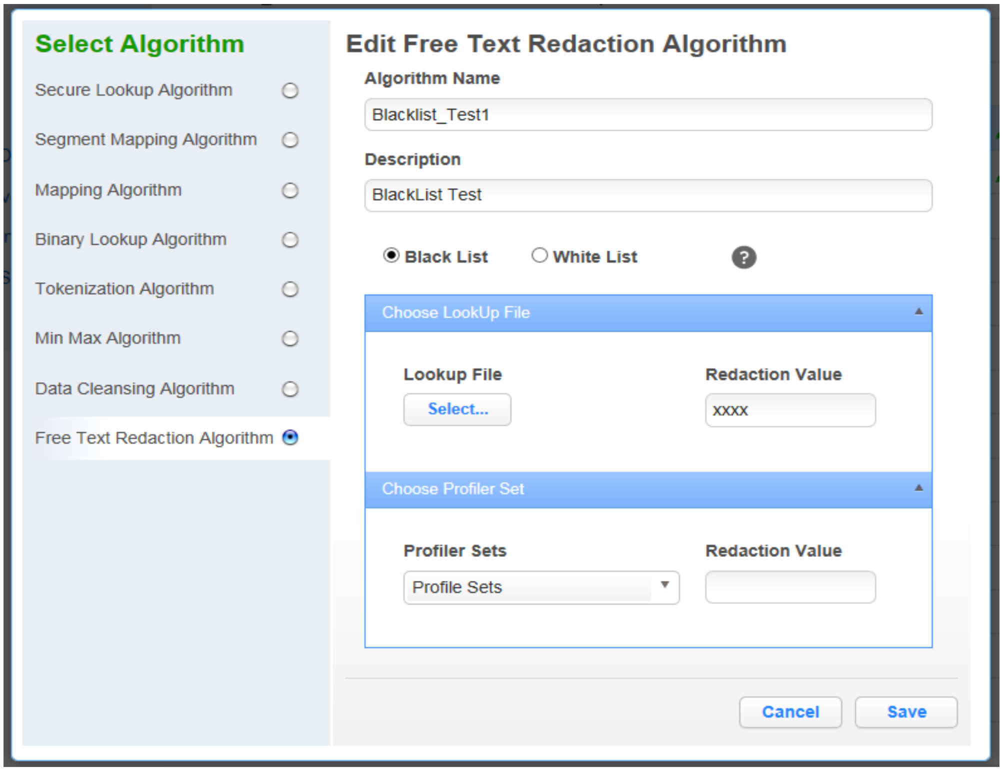
One challenge is that individual words might not be sensitive on their own, but together they can be. The algorithm uses profiler sets to determine what information it needs to mask. You can decide which expressions the algorithm uses to search for material such as addresses. For example, you can set the algorithm to look for “St,” “Cir,” “Blvd,” and other words that suggest an address. You can also use pattern matching to identify potentially sensitive information. For example, a number that takes the form 123-45-6789 is likely to be a Social Security Number.

You can use a free text redaction algorithm to show or hide information by displaying either a “black list” or a “white list.”

Blacklist – Designated material will be redacted (removed). For example, you can set a black list to hide patient names and addresses. The blacklist feature will match the data in the lookup file to the input file.

Whitelist – ONLY designated material will be visible. For example, if a drug company wants to assess how often a particular drug is being prescribed, you can use a white list so that only the name of the drug will appear in the notes. The whitelist feature enables you to mask data using both the lookup file and a profiler set.

For either option, a list of words can be imported from an external text file or alternatively, you can use Profiler Sets to match words based on regular expressions, defined within Profiler Expressions. You can also specify the redaction value that will replace the masked words. Regular expressions defined using Profiler Sets will match individual words within the input text, rather than phrases.


Select Algorithm
Secure Lookup Algorithm ○
Segment Mapping Algorithm ○
Mapping Algorithm ○
Binary Lookup Algorithm ○
Tokenization Algorithm ○
Min Max Algorithm ○
Data Cleansing Algorithm ○
Free Text Redaction Algorithm ●
Edit Free Text Redaction Algorithm
Choose LookUp File
Lookup File
Select...
Choose Profiler Set
Cancel
Save

Algorithm Name: Blacklist_Test1

Description: BlackList Test

● Black List ○ White List

Redaction Value: xxxx

Profiler Sets: Profile Sets

1. Enter **Algorithm Name**.
2. Enter a **Description**.

3. Select the **Black List** or **White List** radio button.
4. Select **Lookup File** and enter **Redaction Value** OR/AND
   Select **Profiler Sets** from the drop-down menu and enter **Redaction Value**.
5. Click **Save**.

Free Text Redaction Example

1. Create Input File.
2. Create input file using notepad. Enter the following text:
   "The customer Bob Jones is satisfied with the terms of the sales agreement. Please call to confirm at 718-223-7896."
3. Save file as txt.
4. Create look up file.
   a. Create a lookup file.
   b. Use notepad to create a txt file and save the file as a TXT. Be sure to hit return after each field. The lookup flat file contains the following data:
      Bob
      Jones
      Agreement

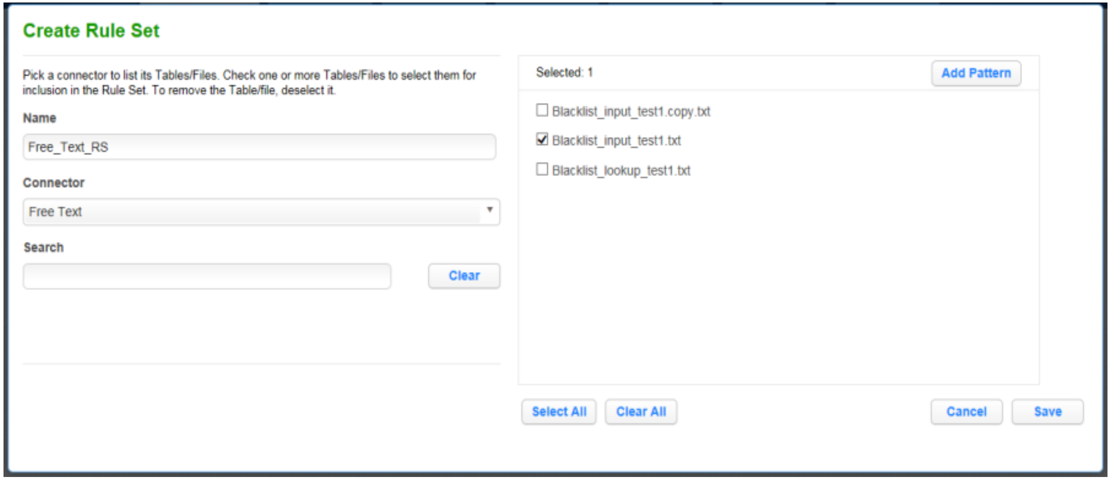
Create an Algorithm

You will be prompted for the following information:

1. For **Algorithm Name**, enter **Blacklist_Test1**.
2. For **Description**, enter **Blacklist Test**.
3. Select the **Black List** radio button.
4. Select **LookUp File**.
5. Enter redaction value **XXXX**.
6. Click **Save**.

Create Rule Set

1. From the job page go to Rule Set and Click **Create Rule Set**.



Create Rule Set

Pick a connector to list its Tables/Files. Check one or more Tables/Files to select them for inclusion in the Rule Set. To remove the Table/file, deselect it.

Name: Free_Text_RS

Connector: Free Text

Search:

Selected: 1

☐ Blacklist_input_test1.copy.txt
☑ Blacklist_input_test1.txt
☐ Blacklist_lookup_test1.txt

2. For **Rule Set Name**, enter **Free_ Text_RS**.
3. From the **Connector** drop-down menu, select **Free Text**.
4. Select the **Input File** by clicking the box next to your input file
5. Click **Save**.

Create Masking Job

1. Use Free_Texr Rule Set
2. Execute Masking job.

The results of the masking job will show the following:

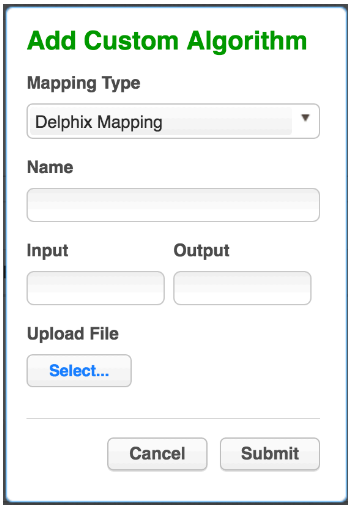
Redacted Input File: The customer xxxx xxxx is satisfied with the terms of the sales xxxx. Please call to confirm at 718-223-7896.

"Bob," "Jones," and "agreement" are redacted.

Adding a New Custom Algorithm

To add a new custom algorithm:

1. Click **Add Algorithm** at the upper right. The Add Custom Algorithm window appears
2. Select a **Mapping Type.**
3. Enter an **Algorithm Name.**
4. Enter values for **Input** and **Output.**
5. Select a **Custom Algorithm File** from the file system.
6. Click **Submit**.

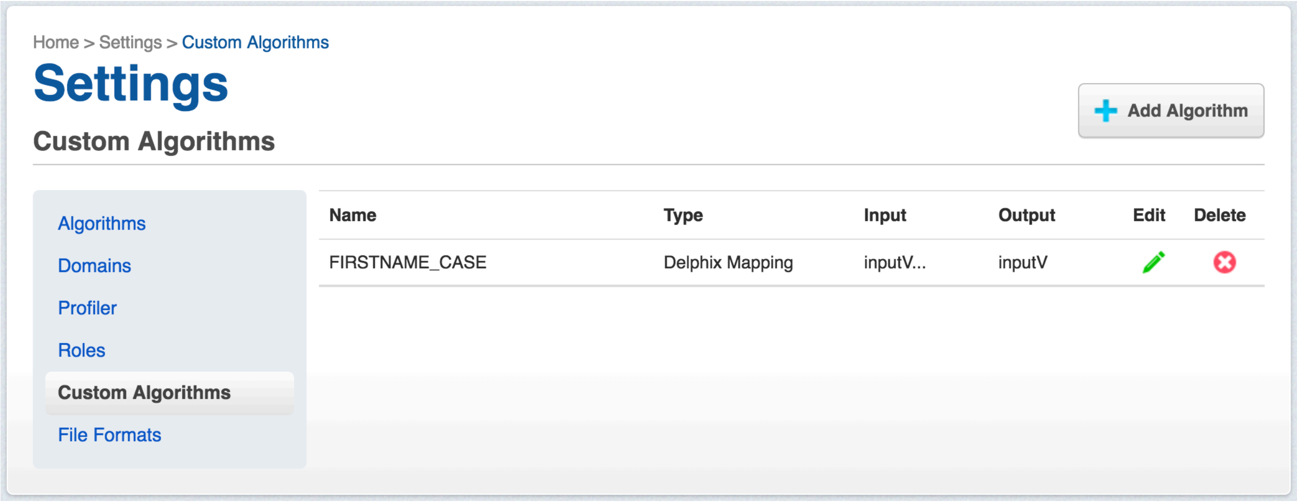


Mapping Type: Delphix Mapping

Related Links

- Managing Custom Algorithms Settings
- Managing Multiple Delphix Masking Engines

Managing Custom Algorithms Settings



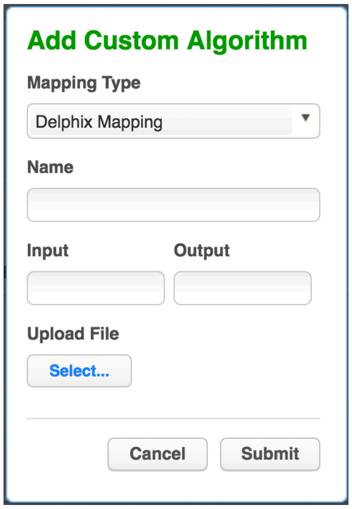
Name	Type	Input	Output	Edit	Delete
FIRSTNAME_CASE	Delphix Mapping	inputV...	inputV	✎	✖

Custom Algorithms Tab

Adding Custom Algorithm

To add a new custom algorithm:

1. Click **Add Algorithm** at the upper right. The Add Custom Algorithm window appears
2. Select a **Mapping Type.**
3. Enter a **Algorithm Name.**
4. Enter values for **Input** and **Output.**
5. Select a **Custom Algorithm File** from the file system.
6. Click **Submit**.



Add Custom Algorithm

Mapping Type: Delphix Mapping

To Default Values/ Best Practices

	Description	Example
Mapplet Name	All CAPS wiht no spaces or special characters	RANDOMDATE
input	string per guidelines, this variable lets mapping talk to Delphix	inputValue
output	string per guidelines, this variable lets mapping talk to Delphix	ouputValue