You can find the most up-to-date technical documentation at:

http://www.delphix.com/support

The Delphix Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to:
feedback@delphix.com.
# Table of Contents

MANAGING ALGORITHM SETTINGS ........................................................................................................ 5
  ALGORITHM SETTINGS TAB .................................................................................................................. 5

ADDITIONAL MASKING ENGINE ALGORITHMS .............................................................................. 6
  PROCEDURE FOR ADDING AN ALGORITHM ...................................................................................... 7

CHOOSING AN ALGORITHM TYPE .................................................................................................... 8
  SECURE LOOKUP ALGORITHM ......................................................................................................... 8
  SEGMENTED MAPPING ...................................................................................................................... 8
    Ignoring or Preserving Specific Values ........................................................................................... 8
  MAPPING ALGORITHM ..................................................................................................................... 9
  BINARY LOOKUP ALGORITHM ....................................................................................................... 9
  TOKENIZATION ALGORITHM ......................................................................................................... 9
  MIN MAX ALGORITHM .................................................................................................................... 9
  DATA CLEANSING ALGORITHM .................................................................................................... 10
  FREE TEXT REDACTION ALGORITHM .......................................................................................... 10

ADDING A SECURE LOOKUP ALGORITHM ..................................................................................... 11

SEGMENTED MAPPING ALGORITHM ................................................................................................. 13
  SEGMENTED MAPPING EXAMPLE .................................................................................................. 13
    To define segments: ....................................................................................................................... 14
    Segmented Mapping Procedure ................................................................................................ 15

MAPPING ALGORITHM ................................................................................................................... 18

BINARY LOOKUP ALGORITHM .......................................................................................................... 20

TOKENIZATION ALGORITHM .......................................................................................................... 20
  CREATING A TOKENIZATION ALGORITHM .................................................................................. 21
  CREATE A DOMAIN ......................................................................................................................... 22
  CREATE A TOKENIZATION ENVIRONMENT .................................................................................... 23
  CREATE AND EXECUTE A TOKENIZATION JOB .............................................................................. 24
  STEPS TO RE-IDENTIFY MASKED INFORMATION ........................................................................ 26
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIN MAX ALGORITHM</td>
<td>27</td>
</tr>
<tr>
<td>PROCEDURE</td>
<td>27</td>
</tr>
<tr>
<td>DATA CLEANSING ALGORITHM</td>
<td>28</td>
</tr>
<tr>
<td>PROCEDURE</td>
<td>28</td>
</tr>
<tr>
<td>FREE TEXT ALGORITHM</td>
<td>29</td>
</tr>
<tr>
<td>FREE TEXT REDACTION EXAMPLE</td>
<td>30</td>
</tr>
</tbody>
</table>
This document provides the steps required to setup agile masking algorithms.

**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 Engine’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**

Within the Settings page, the Algorithm tab displays the Name, Type, and a Description of each algorithm currently available to you. On this tab, you will see the default algorithms and any additional algorithms you have defined. This is also where you can create new algorithms.

**Note:** All algorithm values are stored encrypted. These values are only decrypted during the masking process.
Figure 1 Algorithm Settings Tab

Adding New Masking Engine Algorithms

If none of the default algorithms meet your needs, you can create a new algorithm directly on the Algorithm tab of the Settings page. Then, you can immediately propagate it. Anyone in your organization who has the Masking Engine can then access the information.

**Note:** User-defined algorithms can be accessed by all users and updated by the user who created the algorithm. System-defined algorithms can only be updated by administrators.
Procedure for Adding an Algorithm

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

![Select Algorithm Type Popup](image)

**Figure 2 Select Algorithm Type Popup**

2. Choose one of the following algorithm types. For use examples of when you might want to use each of these algorithm types, see the section Choosing an Algorithm Type below.
   - Secure Lookup Algorithm
   - Segmented Mapping Algorithm
   - Mapping Algorithm
   - Binary Lookup Algorithm
   - Tokenization Algorithm
   - Min Max Algorithm
   - Data Cleansing Algorithm
   - Free Text Redaction Algorithm
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 segmented mapping, described below.

Segmented Mapping

Segmented mapping produces no overlaps or repetitions in the masked data. You can mask up to a maximum of 36 values using segmented 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. You can set the algorithm to produce alphanumeric results (letters and numbers) or only numbers.

Ignoring or Preserving Specific Values in Segmented Mapping

With segmented 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.

**Mapping Algorithm**

A mapping algorithm allows you to state what values will replace the original data. 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.

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

**Binary Lookup Algorithm**

A binary lookup 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.

**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 Delphix Engine stores both the token and the original so that you can reverse masking later.

**Min Max Algorithm**

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.

**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.”

**Free Text Redaction 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 blacklist 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 profile set.

### Adding 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 Rule pane appears.

#### Figure 3 Create Secure Lookup Rule Pane

3. Enter a **Rule Name**. This name must be unique.
4. Enter a **Description**.
5. Select a **Lookup File**.

This file is a single list of values. It does not require a header. Ensure that there are no spaces or returns at the end of the last line in the file.

The following is sample file content:
Smallville
Clarkville
Farmville
Townville
Cityname
Citytown
Towneaster

The Delphix Masking Engine only supports lookup files saved in ASCII or UTF-8 format. If the lookup file contains foreign alphabet characters, you must save the file in UTF-8 format for the Masking Engine to read the Unicode text correctly.

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.

**Adding a New Domain**

1. At the top of the **Domains** tab, click **Add Domain**. A new domain will be created in-line.

2. Enter the new **Domain Name**. The domain name you specify will appear as a menu option on the Inventory screen elsewhere. Domain names must be unique.

3. Select the **Classification** – for example, customer-facing data, employee data, or company data.
4. Select a default **Masking Algorithm** for the new domain.

5. For information about algorithm settings, see Managing Algorithm Settings.

6. Click **Save**.

**Segmented Mapping Algorithm**

Segmented mapping algorithms let you create unique masked values by dividing a target value into separate segments and masking each segment individually. Optionally, you can preserve the semantically rich part of a value while providing a unique value for the remainder. This is especially useful for primary keys or columns that need to be unique because they are part of a unique index.

**NOTE:** When using segmented mapping algorithms for primary and foreign keys, you must use the same segmented mapping algorithm for each key to make sure they match.

**Segmented Mapping Example**

When masking an account number, you can separate it into segments, preserving some segments and replacing others. For example, with the account number NM831026-04:

- **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.
To define segments:

1. For **Number of Segments**, select 3. 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 the sample value) to always be 98 or 99:
   a. For **Segment 1**, select **Type > Numeric**.
   b. Select **Length > 2**.
   c. For **Mask Values Range#**, enter 98,99.

4. Define the next four-digit segment (“1026” in sample value):
   a. For **Segment 2**, select **Type > Numeric**.
   b. Select **Length > 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. Select **Length > 2**.
   c. For **Mask Values Min#**, enter 1.
   d. For **Mask Values Max#**, enter 77.
Using this algorithm, the sample value NM831026-04 might be masked to NM981291-77.

**Segmented Mapping Procedure**

To add a segmented mapping algorithm:

1. In the upper right-hand corner of the **Algorithm** tab, click **Add Algorithm**.
2. Select **Segmented Mapping Algorithm**.

   The **Create Segment Mapping** pane appears.

3. Enter a **Rule Name**.

---

*Figure 4 Create Segment Mapping Pane*
4. Enter a **Description**.

5. From the **Number of Segments** dropdown, select how many segments you want to mask. Do NOT count 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, select the **Type** of segment from the dropdown: **Numeric** or **Alpha-Numeric**.

   **IMPORTANT:** “Numeric segments” are masked as whole segments. “Alphanumeric segments” are masked by individual character.

7. For each segment, choose the Length of the segment (number of characters) from the dropdown (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.)

   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.

   **Note:** 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.

   - **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.
• **Alpha-Numeric** 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.

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

10. To ignore specific characters, enter one or more characters in the **Ignore Character List** box. Separate values with a comma.

11. To ignore the comma character (,), select the **Ignore comma (,)** check box.

12. To ignore control characters, select **Add Control Characters**.

   The **Add Control Characters** window appears.
13. Select the individual control characters that you would like to ignore, or click **Select All** or **Select None**.

14. When you are finished, click **Save**. You are returned to the **Segmented Mapping** pane.

13. **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**.

   If you need additional value fields, click **Add**.

14. When you are finished, click **Save**.

15. 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 Profiler to create your inventory, you do not need to associate the algorithm with a domain. See **Adding New Domains elsewhere in Delphix documentation**.

### Mapping Algorithm

A mapping algorithm sequentially maps original data values to masked values that are pre-populated to a lookup table through the Masking Engine user interface. With the mapping algorithm, 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 10000 unique values in the column you are masking you must give the mapping algorithm **at least** 10000 values.

To add a mapping algorithm:

1. In the upper right-hand corner of the **Algorithm** tab, click **Add Algorithm**.
2. Choose **Mapping Algorithm**.
   
   The **Create Mapping Rule** 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 there’s 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.
   
   To ignore the comma character (,), select the **Ignore comma (,)** check box.

7. When you are finished, click **Save**.
8. 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 Profiler to create your inventory, you do not need to associate the algorithm with a domain. See Adding New Domains elsewhere in Delphix documentation.

**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 is useful for masking binary columns (e.g. blob, image, varbinary, etc).

**To add a binary lookup algorithm:**

1. Click **Add Algorithm** at the top right of the Algorithm tab.
2. Choose **Binary Lookup Algorithm**.
   
   The Create Binary SL Rule pane appears.

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

**Tokenization Algorithm**

Tokenization uses reversible algorithms so that the data can be returned to its original state. Actual data, such as names and addresses, are converted into tokens that have similar properties to the original data (text, length, etc) but no longer convey any meaning.
Here is a snapshot of the data before and after Tokenization to give you an idea of what it will look like.

**Before Tokenization**

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ID, fname, address, ssn</td>
<td>2</td>
<td>Erasmus, 245 Park Ave, 123-45-6789</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>Jason, 45 Omega Suites, 123-45-6789</td>
<td>5</td>
<td>Waldeve, 1 Pulitzer way, 123-45-6789</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Salathiel, 245 Park Ave, 123-45-6789</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**After Tokenization**

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ID, fname, address, ssn</td>
<td>2</td>
<td>Erasmus, L1kgrFFRzafOTUqfpZAmIC==, 123-45-6789</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>Jason, C4v5jr1mKEhKC3acnQKqEk==, 123-45-6789</td>
<td>5</td>
<td>Waldeve, v89pB9b9QISxyYvs/agYUg==, 123-45-6789</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Salathiel, yrLNEnhI8j40ld7y7dXRqwY==, 123-45-6789</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Creating a Tokenization Algorithm**

1. From the Home page, click **Settings**.
2. Click Add Algorithm. You will see the popup below:

3. Select Tokenization Algorithm.
4. Enter a name and description.
5. Click Save.

Create a Domain
After you have created an algorithm, you must associate it with a domain.

1. From the Home page, click Settings.
2. Select Domains.
3. Click Add Domain. You will see the popup below:
4. Enter a domain name and associate it with your algorithm.

Create a Tokenization Environment
1. From the Home page, click the Environments tab.
2. Click Add Environment. You will see the popup below:

3. Select Tokenize/Re-Identify as the purpose.
4. Click **Save**. Note: This environment will be used to re-identify your data when required.

At this point, you can proceed in the same fashion as if you were using Delphix to perform normal masking. You have made all the changes needed to use Tokenization (reversible) algorithms instead of Masking (irreversible) algorithms. Note it is possible to create two different environments for the same application – one for masking and one for tokenization.

**Create and Execute a Tokenization Job**

1. From the Home page, click **Environments**.
2. Click **Tokenize**.
3. Set up a Tokenize job using tokenization method. Execute the job.

4. You will be prompted for the following information:

![Create Tokenization Job](image)
a. **Job Name** — A free-form name for the job you are creating.
b. **Tokenization Method** — Select Tokenization Method.
c. **Multi Tenant** — Check box if the job is for a multi-tenant database.
d. **Rule Set** — Select a rule set that this job will execute against.
e. **Generator**
f. **No. of Streams** — The number of parallel streams to use when running the jobs. For example, you can select two streams to run two tables in the ruleset concurrently in the job instead of one table at a time. (This option only appears if you select DMsuite as the Generator.)
g. **Remote Server** — (optional) The remote server that will execute the jobs. This option lets you choose to execute jobs on a remote server, rather than on the local Delphix instance. Note: This is an add-on feature for Delphix Standard Edition. (This option only appears if you select DMsuite as the Generator.)
h. **Min Memory (MB)** — (optional) Minimum amount of memory to allocate for the job, in megabytes. (This option only appears if you select DMsuite as the Generator.)
i. **Max memory (MB)** — (optional) Maximum amount of memory to allocate for the job, in megabytes. (This option only appears if you select DMsuite as the Generator.)
j. **Commit Size** — (optional) The number of rows to process before issuing a commit to the database.
k. **Feedback Size** — (optional) The number of rows to process before writing a message to the logs. Set this parameter to the appropriate level of detail required for monitoring your job. For example, if you set this number significantly higher than the actual number of rows in a job, the progress for that job will only show 0 or 100%.
l. **Disable Constraint** — (optional) Whether to automatically disable database constraints. The default is for this check box to be clear and therefore not perform automatic disabling of constraints. For more information about database constraints, see Enabling and Disabling Database Constraints.
m. **Batch Update** — (optional) Enable or disable use of a batch for updates. A job's statements can either be executed individually, or can be put in a batch file and executed at once, which is faster.
n. **Truncate** — (optional) Whether to truncate target tables before loading them with data. If this box is selected, the tables will be "cleared" before the operation. If this box is clear, data is appended to tables, which potentially can cause primary key violations. This box is clear by default.

o. **Disable Trigger** — (optional) Whether to automatically disable database triggers. The default is for this check box to be clear and therefore not perform automatic disabling of triggers.

p. **Drop Index** — (optional) Whether to automatically drop indexes on columns which are being masked and automatically re-create the index when the masking job is completed. The default is for this check box to be clear and therefore not perform automatic dropping of indexes.

q. **Prescript** — (optional) Specify the full pathname of a file containing SQL statements to be run before the job starts, or click **Browse** to specify a file. If you are editing the job and a prescript file is already specified, you can click the **Delete** button to remove the file. (The Delete button only appears if a prescript file was already specified.) For information about creating your own prescript files, see [Creating SQL Statements to Run Before and After Jobs](#).

r. **Postscript** — (optional) Specify the full pathname of a file containing SQL statements to be run after the job finishes, or click **Browse** to specify a file. If you are editing the job and a postscript file is already specified, you can click the **Delete** button to remove the file. (The Delete button only appears if a postscript file was already specified.) For information about creating your own postscript files, see [Creating SQL Statements to Run Before and After Jobs](#).

s. **Comments** — (optional) Add comments related to this provisioning job.

t. **Email** — (optional) Add e-mail address(es) to which to send status messages.

5. When you are finished, click **Save**.

**Steps to Re-Identify Masked Information**

Use the Tokenize/Re-Identify environment.

1. From the Home page, click **Environments**.
2. Click **Re-Identify**.
3. Create a re-Identify job and execute.
Min Max Algorithm

This algorithm allows you to make sure all the values in the database are within a specified range.

![Algorithm Selection](image)

**Procedure**

1. Enter an **Algorithm Name**.
2. Enter a **Description**.
3. Enter **Min value** and **Max value**.
   For example, if you want all ages to be masked to something 18 years old or younger, enter Min Value 0 and Max Value 18.
4. Click **Out of range Replacement Value**.
   If “Out of range Replacement value” is checked, the masking engine will use a default value when in cannot evaluate the input.
5. Click **Save**.
Data Cleansing Algorithm

If the target data needs to be put in a standard format prior to masking, you can use these algorithms.

Procedure

1. Enter Algorithm Name.
2. Enter a Description.
3. Select Lookup file location.
4. By default, delimiter separating values is an equals sign (=). If you prefer, you can change this to another symbol, such as an asterisk (*).
5. Click Save.

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

NYC=NY
NY City =NY
New York=NY
Manhattan=NY
Free Text Algorithm

This section provides an overview of how to create free-text redaction algorithms. For more in-depth information, see the separate guide on this topic.

You will be prompted for the following information:

1. Enter an Algorithm Name.
2. Enter a Description.
3. Select the Black List or White List radio button.
4. Select look up file and enter redaction value
   OR/AND
5. Select Profiler Set from drop down and enter redaction value
6. Click Save
Free Text Redaction Example

1. Create input file.
   a. 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.”
   b. Save file as txt.

2. Create a lookup file.
   a. 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

3. Create the algorithm. You will be prompted for the following information:
   a. Enter the algorithm name: Blacklist_Test1.
   b. Enter a description – Blacklist Test
   c. Select the Black List radio button
   d. Select the look up file and enter redaction value “XXXX.”
   e. Click Save.

4. Create a Ruleset:
   a. From the Environments page, click the Rule Set tab.
   b. Click Create Rule Set. You will be prompted for the following information:
      i. Enter Rule Set Name: Free_Text_RS
ii. From the Connector Mode drop-down menu, select Free Text.

iii. Select the Input File by clicking the box next to your input file.

iv. Click Save.

5. Create the masking job. When you do:

   a. Use the Free_Text Rule Set that you created in step 4. On the Environments page, click Mask.

   b. Execute Masking job.

      i. On the Environments page, click the name of the job.

      ii. Click Action.

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.

The words “Bob,” “Jones,” and “agreement” are redacted.