Data Masking for HIPAA Compliance

The Safe Harbor Method: Practical Implementation Techniques

Abstract

The Health Insurance Portability and Accountability Act of 1996 (HIPAA) Privacy Rule mandates the de-identification of specific types of Protected Health Information (PHI) for covered entities and their business associates.

This paper discusses each of the items listed in §164.514(b) of the Privacy Rule's Safe Harbor de-identification standard and illustrates various data masking techniques which can assist in satisfying that requirement.

Implementation Details

This paper is specifically focused on the practical issues and techniques involved in the preparation of de-identified HIPAA compliant databases. Specific worked examples of data masking practices which can assist with meeting the mandated Safe Harbor data de-identification requirements are provided. It should be noted that no discussion of the alternate Expert Determination mode of meeting the HIPAA standard is provided in this paper.

The authors of this paper offer a data masking software solution called Data Masker and it is used to illustrate the example masking techniques. If there are questions or issues regarding the structure of your data and how it might be de-identified which are not addressed in this paper we would be pleased to provide a case-specific example.

The worked examples in this paper are loosely based on the HIPAA data de-identification guidance provided on the Office for Civil Rights (OCR) website located at

http://www.hhs.gov/ocr/privacy/hipaa/understanding/coveredentities/De-identification/guidance.html

The Data Masker Team
Info@DataMasker.com
http://www.DataMasker.com

Table of Contents

Disclaimer

The contents of this document are for general information purposes only and are not intended to constitute specific professional advice of any description. The provision of this information does not create a business or professional services relationship. Net 2000 Ltd. makes no claim, representation, promise, undertaking or warranty regarding the accuracy, timeliness, completeness, suitability or fitness for any purpose, merchantability or any other aspect of the information contained in this paper, all of which is provided "as is" and "as available" without any warranty of any kind.

Data requiring HIPAA de-identification varies widely in content and structure and each has a unique configuration. Readers should take appropriate professional advice prior to performing any actions.

The Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule

The HIPAA Privacy Rule

The HIPAA Privacy Rule is intended to protect information (however held) which may identify an individual. In general the requirement for inclusion as HIPAA sensitive information is any data, including demographic details, which relates to:

- *the individual's past, present, or future physical or mental health or condition,*
- *the provision of health care to the individual, or*
- *the past, present, or future payment for the provision of health care to the individual, and that identifies the individual or for which there is a reasonable basis to believe can be used to identify the individual. Protected health information includes many common identifiers (e.g., name, address, birth date, Social Security Number) when they can be associated with the health information listed above.*

The determination of which types of data are considered to be PHI data essentially reduces to a determination of whether the data contains any information which may identify an individual and also contains the associated health data content.

Note: the Protected Health Information (PHI) section of the HIPAA Guidance page contains specific examples of what is considered sensitive health data and what is not. Those examples will not be reproduced here. However, there is one specific case which should be commented on:

> *The relationship with health information is fundamental. Identifying information alone, such as personal names, residential addresses, or phone numbers, would not necessarily be designated as PHI. For instance, if such information was reported as part of a publicly accessible data source, such as a phone book, then this information would not be PHI because it is not related to heath data (see above). If such information was listed with health condition, health care provision or payment data, such as an indication that the individual was treated at a certain clinic, then this information would be PHI.*

It is important to remember that the HIPAA Guidance page is specifically concerned with HIPAA compliance. As part of a practical data de-identification strategy you will almost certainly want to ensure that Personally Identifiable Information (PII) is rendered anonymous even if such de-identification is not strictly required under the

Privacy Rule. In other words, a practical de-identification solution for your organization will rarely be limited to just HIPAA compliance.

One of the most common attributes of useful data is the way in which it is interrelated with other data records in the system. The HIPAA Guidance does not offer an opinion on whether the relationships need to be maintained once the data has been anonymized – it just insists that the data, as a collection, cannot be subsequently used to identify an individual. In practice, a de-identification process which destroys the relationships between data records will almost never be satisfactory. Considerable care must be taken when masking the data to change distributed PII data in an identical and synchronized manner.

HIPAA Privacy Rule Data De-Identification Methods

There are two implementation specifications a HIPAA covered entity can follow to meet the Privacy Rule data de-identification standards. These two specifications (called methods) are covered by sections 164.514 parts (b) and (c) of the Privacy Rule and are simply named the Safe Harbor Method and the Expert Determination Method.

The Expert Determination Method basically reduces to a "judgment call" by a suitably knowledgeable and qualified person who, in their professional opinion, believes the operations performed on the data have rendered it into such a state that there is a very small risk that anybody viewing the data could use it to identify an individual. No discussion of the Expert Determination Method is contained within this paper.

The Safe Harbor Method prescribes a list of identifiers related to an individual (or relatives, employers, or household members of the individual) which should have data de-identification operations performed on them.

HIPAA Privacy Rule Safe Harbor Examples

The sections below discuss the identifiers listed under the HIPAA Privacy Rule Safe Harbor Method and provide examples of strategies and techniques which may assist in the data de-identification process for those items while retaining meaningful relationships with other data records. The list of techniques for each identifier is not exhaustive and any one technique may not be appropriate in every specific case. Also, a technique demonstrated for a specific identifier may well be useful for other identifiers even though it is not listed there for space reasons.

Before masking any identifier, perform a thorough check to see if the data it contains is reproduced in other data records in a de-normalized manner. If the data exists elsewhere, then the synchronization techniques discussed in *Social Security Numbers* section below should be considered. Rarely is it possible to synchronize data after it has been de-identified – the related data items must be processed together in order to preserve relationships. Take particular care to look for data which is stored in different formats in different locations. The classic example of this is an SSN which is stored as a number in some locations and text in another and can often be held in differently sized columns.

Names

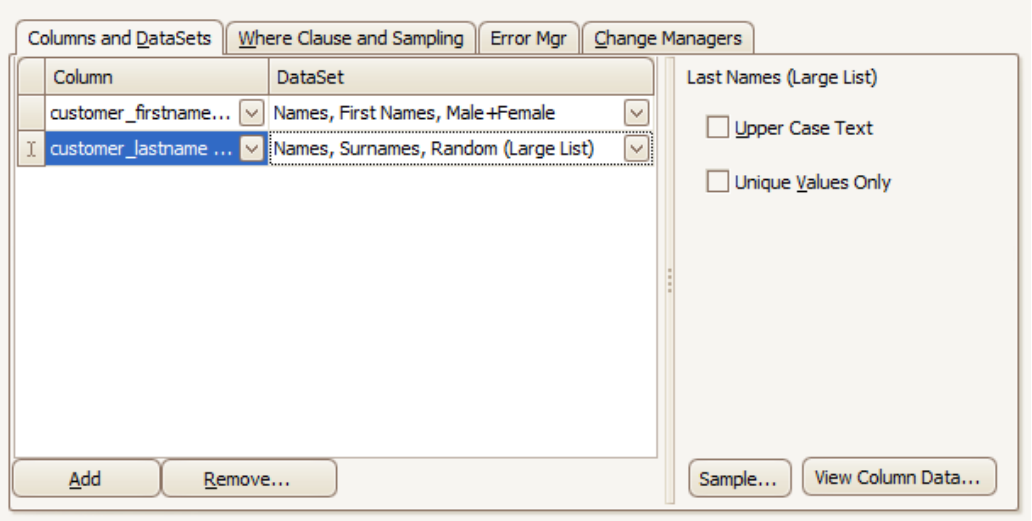
Names, if there is no synchronization required, are usually one of the more straightforward of the Safe Harbor identifiers to de-identify. Typically the preferred technique here is one of substitution from pre-prepared datasets of random names.

One important point to note is that, although it is not strictly necessary for HIPAA PHI data compliance, almost never will end users of the masked data consider it acceptable to have names substituted with replacement data that do not look like names. This extends to the point where forenames need to be substituted with gender appropriate first names.

In some cases it may be desirable to synchronize the masking of family surname records so that all family members have identical surnames. Note that when synchronizing the masking of family surname records so that all family members have identical surnames, it is important to take care that the synchronization is structured so that one-to-one equivalencies are not placed on all values. Thus while members of a specific family named "*Smith*" could all be given the de-identified surname of "*Jones*" another family named "*Smith*" should not automatically be given the name of "*Jones*" as well. Those family members should be given a different name in the masked data – not doing so introduces a global equivalency (all "*Jones*" records are actually "*Smiths*") and this would be incompatible with the HIPAA Privacy Rule specification.

Another thing to consider is that you will need to make decisions on whether to leave NULL or empty values as-is. In many cases, the end users of the data will not appreciate having data added into places where it was previously empty. The reverse can also be true. While the NULL'ing out of data is a valid masking technique, a careful analysis of the data must be performed first in order to ensure that the end users of the data will not need that information. This perseveration of the database demographics (or dynamics) is important since it can directly affect index distribution and performance as well as record expansions within the same, or other, blocks or pages.

The screenshot below shows the simple configuration of a Data Masker Substitution Rule designed to mask the first and last names of two columns in a table. No synchronization is implemented in this particular example and the first name data is not set to be gender specific.



In practice, however, the gender is usually always specified in PII/PHI data. It should be noted that gender is not always as simple as Male (M) or Female (F). Care should be taken to identify all distinct values in the gender column. In medical records particularly it is not unusual to find an "unknown" gender or a defined gender description such as "X", which can mean "Intersex" or "of indeterminate sex". Making the false assumption that gender can only ever be male or female would usually result in the other groups being skipped during the masking process.

This particular omission is known as a "*Where Clause Skip*"and it is discussed in detail in a companion whitepaper entitled "Data Masking: Everything You Need to Know" available from the Data Masker website. We recommend that paper to anyone wishing to mask HIPAA data as it documents many of the tricks and traps associated with data masking. In the specific case of gender, it is usually best to update all of the patients with one gender forename as a first step (for example: female) and then take a second pass through the data replacing the names with male forenames where the gender is specifically identified as being male. This ensures that no names are left unmasked.

Geographic Locations

The Safe Harbour Method states that the following information must be de-identified:

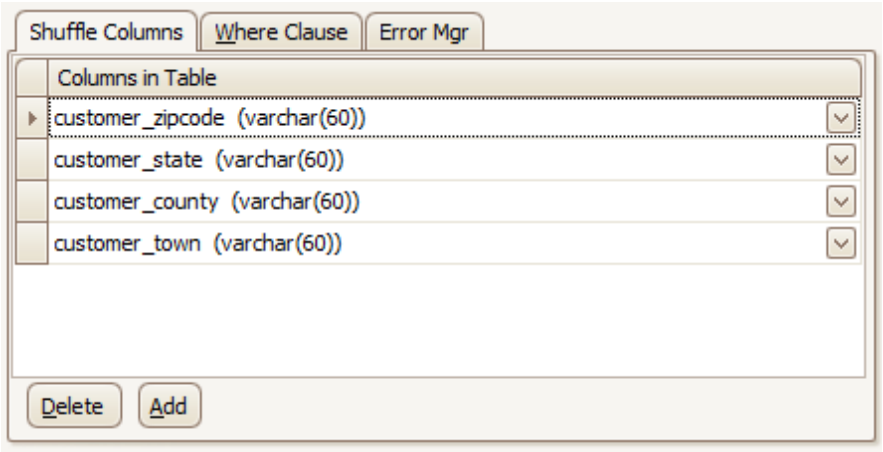
> *All geographic subdivisions smaller than a state, including street address, city, county, precinct, ZIP code, and their equivalent geocodes, except for the initial three digits of the ZIP code*

One key point to note here is that in this case the information to be de-identified typically consists of several fields all of which are related to one another. For example, any one ZIP code points to a specific state, county and town.

A method which may satisfy the PHI data de-identification is simply to remove all geographical information leaving only the first three digits of the zip code (as long as that zip code has greater than 20000 people). The downside is that this makes the data much less realistic for the end users and tends to play havoc with any data validation software which might try to process it during testing.

Shuffling the geographical data, moving the data as a unit out to another row while moving another set of values into its place, disassociates any specific geographical location from the record and preserves the correlation of the values (zip code, state, county and town). This has a drawback in that the output data still indicates that "*a PHI event is associated with this location*". It depends on the meaning of the data – but typically there are usually relatively fewer situations in which such a technique would be compatible with the PHI Privacy Rule.

In the event shuffling is appropriate, the image below illustrates the relevant section of a Data Masker Shuffle Rule designed to distribute a group of the columns within a row randomly to other rows in that same table. The rows in the table receive the same group of data from other rows at random and after shuffling there will be no duplication of rows or unshuffled rows and all of the zip, state, county and town information will move from row to row together.

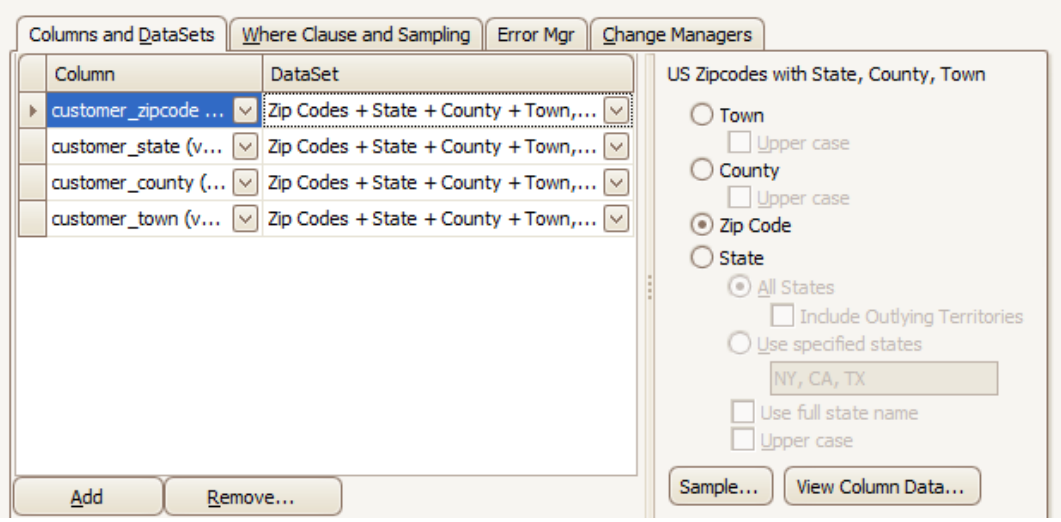


One technique which may be more applicable to the Geographic Location requirement is the replacement of geographical information with similar, real, values. For example, a data record with a zip code, state, county and town field could be de-identified by

replacing the data by a random zip code, state, county and town. The technique here is that for any one record the substitution values are correlated. The replacement zip code really does reference the replacement state, county and town. Since any given zip code, state, county and town is public information and they are randomly selected, a substitution of this type may well be appropriate for PHI data de-identification.

There may well be cases in which the geographic distribution of the original data is required to be preserved in the de-identified data. A common requirement is that a geographic location be replaced with a geographic location from within the same state. A finer granularity (replacing within town or street) would seem to preserve too much information to be Privacy Rule compliant. In such situations, a de-identification process based on the Expert Determination Method may be more appropriate.

The screenshot below shows a section of a Data Masker Substitution Rule designed to mask the zip code, state, county and town columns of a table with correlated values from a dedicated dataset.



Dates

With dates, the only element not required to be de-identified is the year value. Even then, all elements of a date indicative of an age greater than 89 must be de-identified. This includes all date information directly related to an individual such as birth date, admission date, discharge date, death date.

Rarely is it useful to just replace dates with random values. Dates almost always have a relationship and a meaning in context with other dates. Simply randomizing dates in isolation can introduce a large number of discrepancies into the data. For example, a date of admission can be moved ahead of a date of discharge.

There are variations on the theme of simple date replacement which have more utility – one such technique is date variance. In date variance each existing date is used as a base and a random change (typically an integer number of days) is applied to that date. In this way the dates are changed to a non-identifiable value but they do not move too far from the actual date – this can assist in ensuring the masked dates remain meaningful and that the age demographic of the database is not affected by the data masking actions.

However there are problems with this technique. Even if the applied variance is not negative it is still possible to move dates out of sequence. For example, if a date of admission and a date of discharge are 1 day apart and a variance of +12 days is applied to the date of admission and a variance of +3 days is applied to the date of discharge then the masked date of discharge will be in the past with respect to the date of admission. In addition, unless care is taken with the date variance technique it is possible to move dates into the future. In other words, if a date in a medical record is only 12 hours old and a variance of +14 days is applied then the masked date in the record is actually a date two weeks in the future.

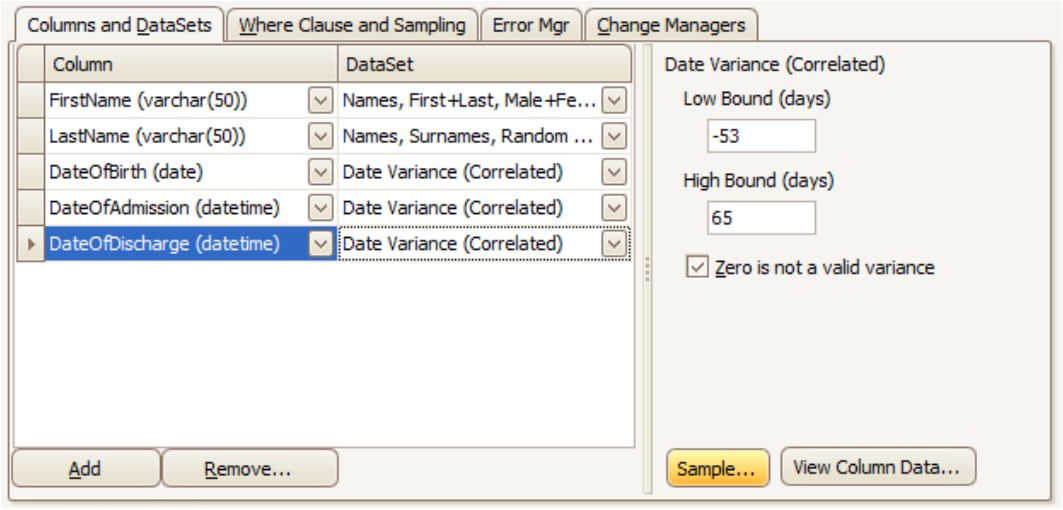
The Data Masker software provides facilities for both simple date substitution and date variance for use in situations where such techniques are appropriate and they will not be illustrated here as they are fairly easy to implement.

A synchronized date variance approach can be used to render dates anonymous while still preserving their relationship to other dates. In this method a random date variance is chosen and then applied to all related dates. For example, a record may get a date variance of +8 days assigned to it and all dates associated with that record are then adjusted by 8 days. A second record may get a date variance of -2 days and all associated records then have two days subtracted from them. No dates remain the same – but the correlation and relationships in any one group of related dates changes in a synchronized manner.

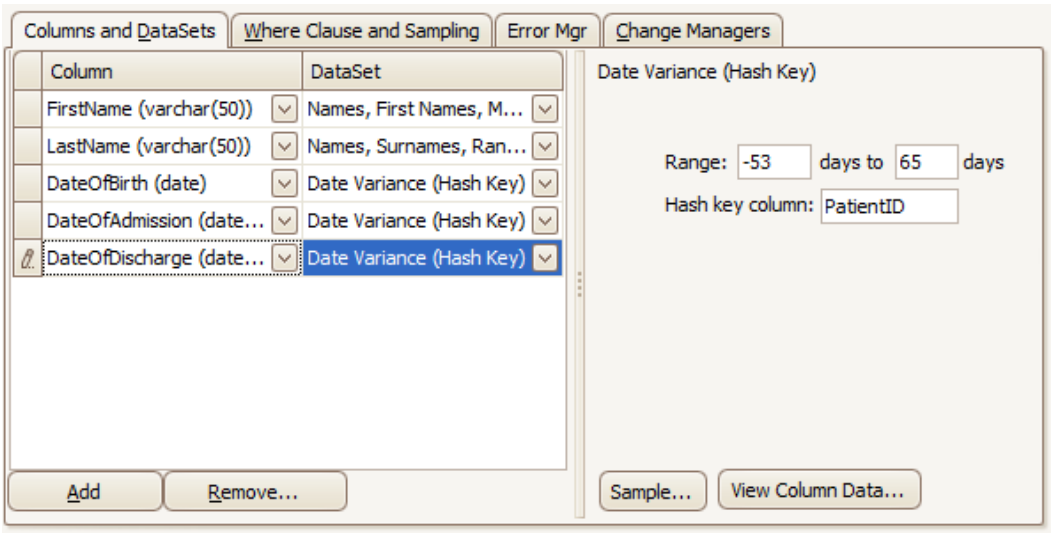
In general, there are two situations in which related dates present themselves. In the first, simpler, case all dates are present in the same row of the same table. For example a data record in a table which contains the fields

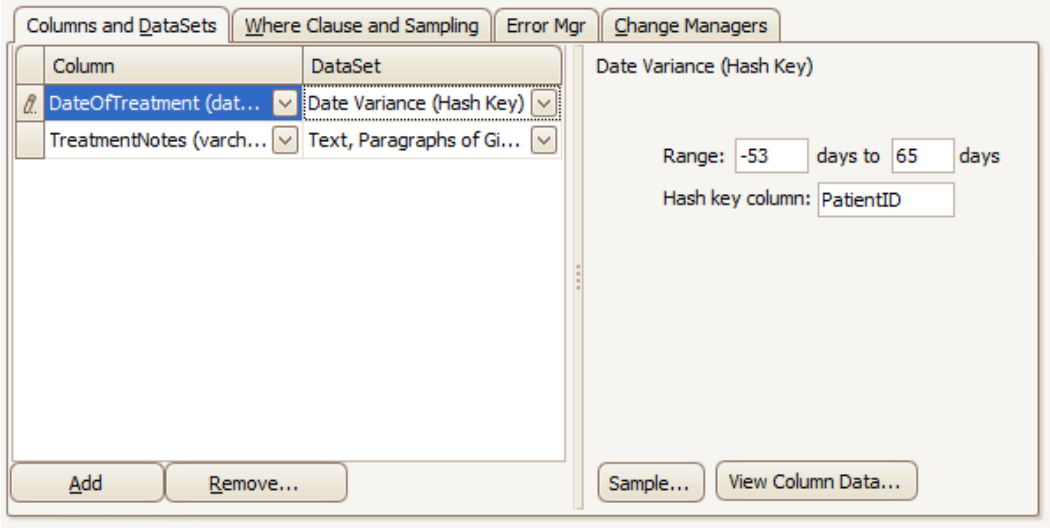
```
FirstName|LastName|DateOfBirth|DateOfAdmission|DateOfDischarge
```


In such a situation a Data Masker Substitution Rule could be used in conjunction with the Correlated Date Variance dataset. This special dataset will apply a date variance within the same record to all configured values in the masking rule. The image below shows a section of a rule in which each record in a table will be processed and the *DateOfBirth*, *DateOfAdmission*, and *DateOfDischarge* fields will all receive a random non-zero variance between -53 days and +65 days. Because the correlated dataset is used, within any one row of the table, the same variance will be applied to each field but each row will receive a random variance. This achieves the goal of dates within the same row which are rendered anonymous yet synchronised to remain meaningful in relation to each other. In the example below, the first and last names are also masked with suitable relevant replacement data. This is just done for efficiency to avoid the need to make a second pass down the table with another rule.



If the dates to be masked are not within the same row of a table then a different technique must be used if they are to be modified in a synchronized way. To achieve this result, the Data Masker software provides a Hash Key Date Variance dataset. The Hash Key Date Variance dataset will use a common field in the record to generate a specific date variance for that record. The examples below show samples of two separate Data Masker Substitution Rules. Both of these rule types mask one or more dates in the table in a synchronised way using the *PatientID* field as the key.





In the above examples, the top masking rule is designed to operate on a table named *Patient* and the second on a table named *Treatment*. Related records in both tables have the same value for the *PatientID*. Accordingly this common column is configured in all of the Hash Key Date Variance datasets and it will be used to generate, for any one distinct *PatientID*, a random non-zero date variance value between -53 days and +65 days.

Using this technique, a record with the same *PatientID*, anywhere in the database, will be able to have its dates adjusted in a manner which is synchronized with all related records. In other words, the variance for any one set of related records will be a single bounded random value and other groups of related records will receive other random variances. This procedure can be extended across any number of tables and date fields. It should be noted that it does not matter to the Hash Key Date Variance dataset if the *PatientID* is a number in one table and a character in another since such storage differences are handled transparently.

Also note that the variance will not necessarily be the same each time the rule is run. Modifying data over repeated runs to the same random values is a technique known as consistent masking and, while it is possible to achieve, it is something of an advanced topic and will not be discussed here. If you need distributed, consistent masking please contact us for implementation advice and also a thorough discussion of the techniques advantages and disadvantages.

The Safe Harbor Method also specifies that all elements of a date indicative of an age greater than 89 must be de-identified. There are various ways of implementing this specific requirement. Usually a two pass approach is taken. All dates are anonymized and then a second set of rules making another pass through the data and operating only on specifically selected records representing patients older than age 89 are used to modify those records back down below the upper limit age.

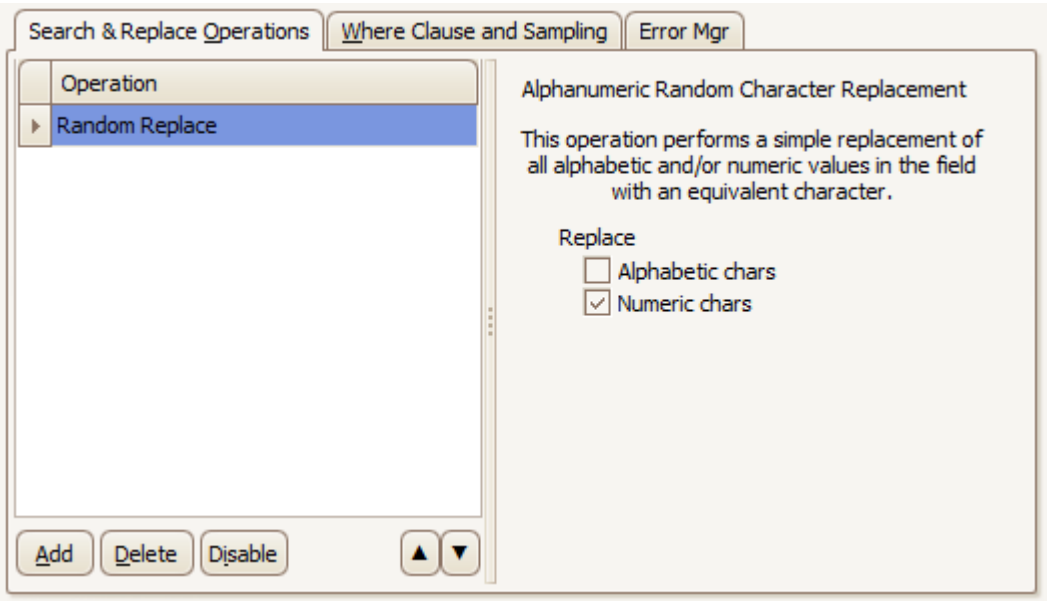
Telephone Numbers, Fax numbers

The difficulty in masking data items such as telephone numbers and fax numbers lies not so much in their replacement with masked data, rather, it is in the finding of them within text fields and sometimes also in the preservation of the multitude of formats in which they can be found.

In regards to the first case, a database may well have free format textual notes associated with a record. This field could contain phone numbers along with other personal information. In such cases, the field is typically just NULL'ed out – but there is another option – one in which the data can be replaced by random characters. This preserves the look and feel of the text data without leaving any PII information in place. End users of the masked database (testers, developers) will see correctly formed content where they expect to see it and this makes their job considerably easier.

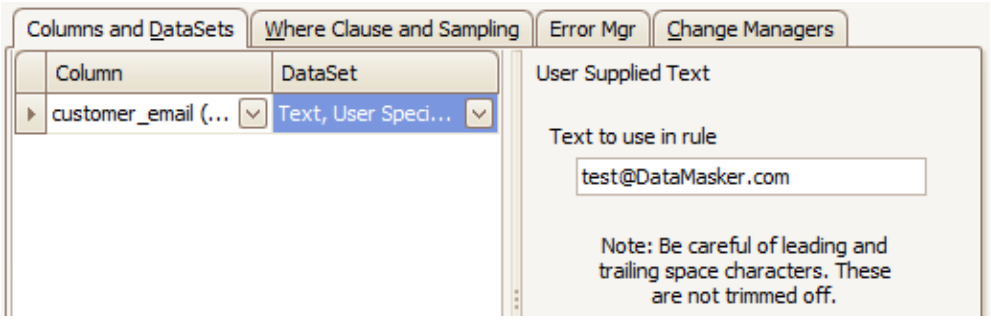
In the second case, even if not embedded in other data a telephone number may be represented in a variety of formats: with and without spacing, with and without brackets, with and without area codes and potentially other seemingly endless variations. If there is variation in the storage format of the telephone numbers in the database and they are all replaced with a random (or constant) number in a distinct format *(123) 456-7890* then the data will have been "*cleaned up*" and hence will not be truly representative of the original data content. If the end use of the database is such that this is not an issue, then the Data Masker Telephone Numbers dataset can be used and the masking operation is easy to set up.

In the event that it is desirable to preserve the format of the numbers a Data Masker Search and Replace Rule can be used in conjunction with the Random Replace Operation to substitute all number digits in a field. The screen shot below shows a Search and Replace Rule configured to run down a *Customer_Telephone* column in a table and replace the telephone numbers it contains with randomized values. This will preserve the formatting and punctuation associated with all numbers while still rendering them anonymous. If the column was in free format containing text and numbers, the "*Alphabetic Chars*" option could also be checked and in addition to the replacement of numeric digits, all letters will be replaced with case sensitive random alternates.

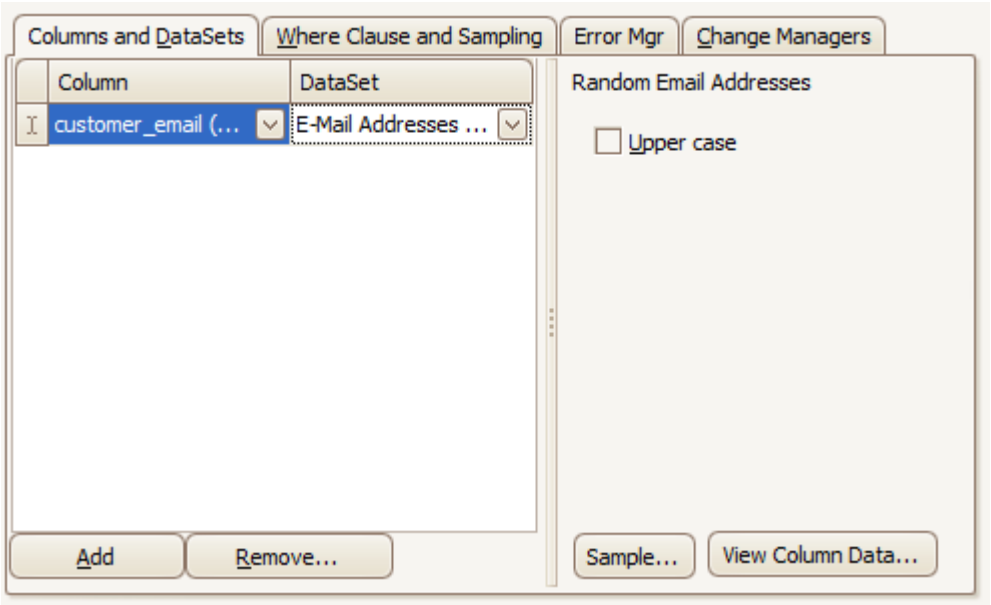


Email Addresses

When masking email addresses one question which must be asked is whether the systems which use and process the data are capable of generating emails either automatically or on demand. In such an environment, it is imperative that the masked email values never correlate to a real email address lest a test run of a process generate real email messages. For this particular type of requirement, replacement with a constant dummy email address using the sites own domain is usually used although nulling the email addresses out entirely is sometimes a valid option. The image below shows a Substitution Rule in which all email addresses in a column are replaced by a constant dummy email address which loops back to the sites own mailbox.



For situations in which substitution with random values is applicable, a dataset of randomly generated email addresses can be used. This dataset mimics the variety of address sizes and domain names commonly found in live data. The image below shows a Data Masker Substitution Rule configured to mask an email address column in a table.



As with telephone numbers, be aware that email addresses can often be imbedded in free format text fields such as notes, or memos. In that event, a Data Masker Search and Replace Rule using a RegEx Replace Operation could mask the email addresses inside the content while leaving the remainder of the text to be masked by a second pass with a different technique. The later section on *Web Universal Resource Locators (URLs)* discusses this technique in considerable detail.

Social Security Numbers

There are several issues with Social Security Numbers (SSN's) which must be considered when rendering them anonymous.

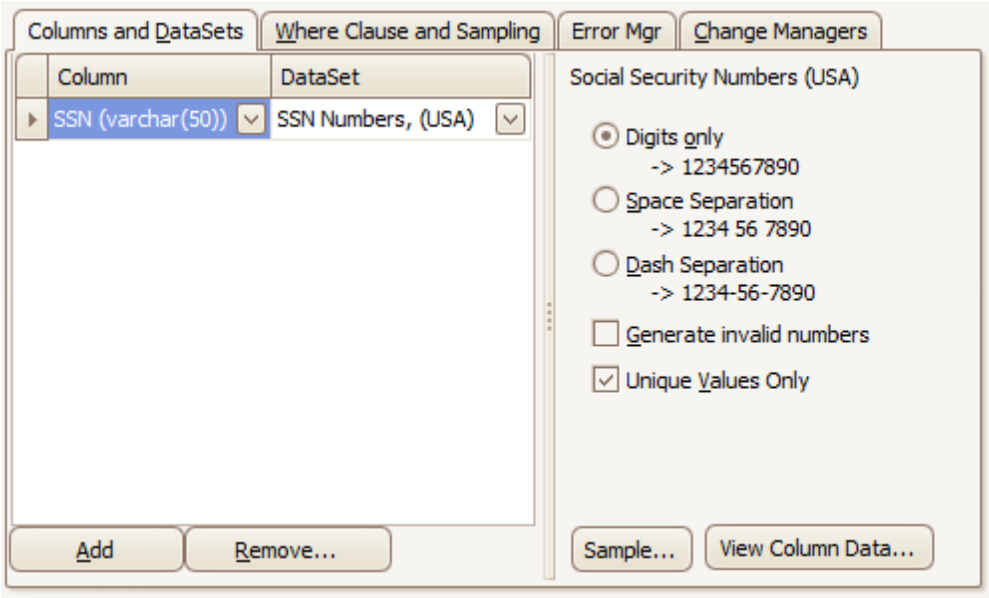
There are known (and publically documented) ranges of SSN numbers which have been issued and others which have not yet been issued. In many systems if valid SSNs are replaced with random invalid SSN's, the presence of that invalid SSN may cause validation routines to fail. For example, a front end screen on a test database may refuse to accept the update of a telephone number because it also attempts to validate the SSN during the update process. In such cases, invalid SSN numbers cannot be used and SSN numbers must be replaced with randomly generated valid numbers. In some circumstances, it is possible that all SSN's can be replaced with a known, constant, dummy SSN.

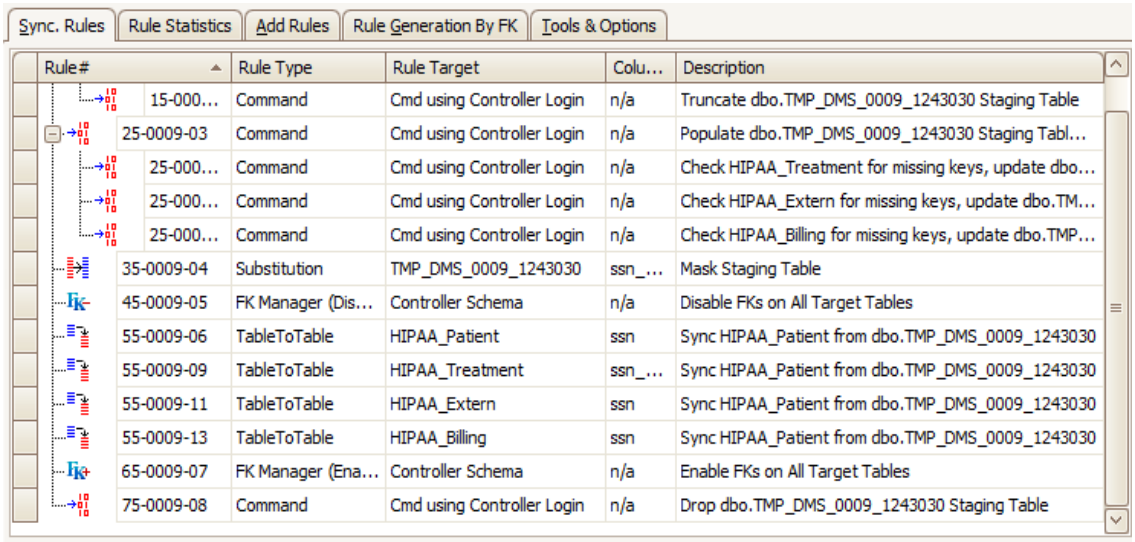
SSN numbers are unique to an individual, when generating them randomly, care must be taken to ensure that no duplicates are created.

For any one individual, it is quite common to see SSN numbers stored in multiple locations throughout the database. In such circumstances, a synchronized change of all SSN numbers is usually desirable. This means that if an SSN of *1234567890* changes to *2335534546* in one table then all occurrences of the 1*234567890* number must also change to that exact same new value.

As an additional complication, since the SSN uniquely identifies an individual, it is sometimes used as a join condition between related records. If that is the case, it is rarely advisable to randomly replace SSN values with new values since this will destroy the relationships between related data and render the resulting database unusable. With the Data Masker software synchronised update functionality is quite simple to achieve even if the item to be masked is used as a join key.

The screen shots below show a Substitution Rule on a single table using the SSN dataset and the second screenshot shows that Substitution Rule embedded in a Synchronisation Manager Rule which will cascade the change out to two other tables. It is possible to cascade to any number of tables whether a Foreign/Parent key enforces the relationship or it is implied at the application level. In addition, it is also possible to cascade such synchronised changes out to tables in other databases or schemas – although such a distributed cascade is something of an advanced topic and will not be discussed here. If your synchronization requirements encompass distributed databases please contact the Data Masker Team for implementation advice. It should be noted that distributed synchronization operations are possible even if the remote databases are of different versions and types.



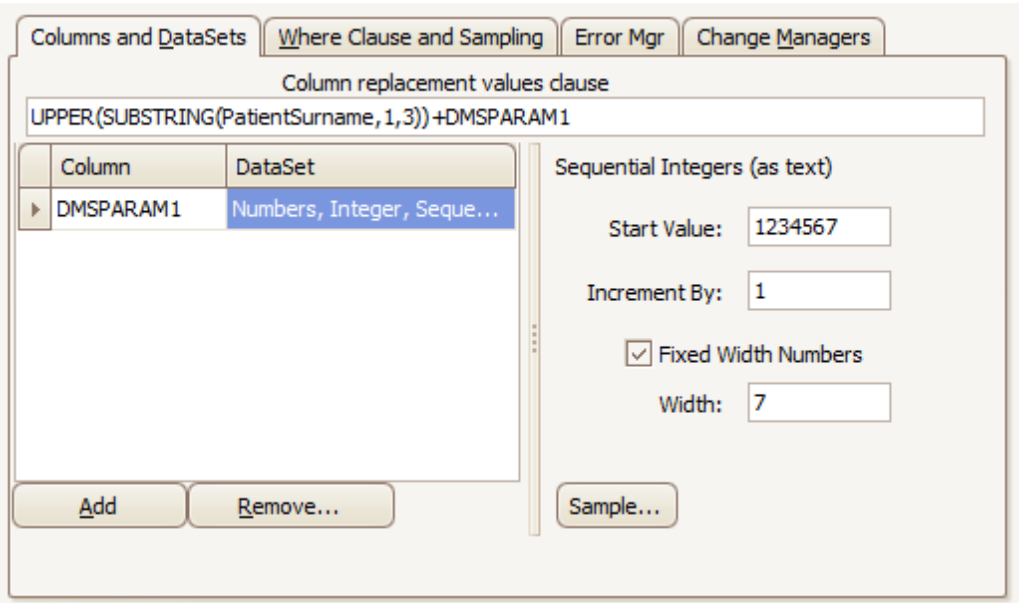


Numbers (Medical Record, Plan Beneficiary, Account)

In many cases numbers of this type are not simple randomly generated or sequential values. Sometimes they are what is known as "intelligent keys" and contain other information from the record. The classic example of this is a record number that uses the first three characters of a surname plus other randomly assigned digits. For example, a patient with a surname of "*Smith*" might have a medical record number like *SMI2423534*.

Clearly there is a requirement to render both the name and the medical record number anonymous. However if the patients name is changed to "*Jones*" it would be expected that the new medical record number will start with the characters "*JON*".

Rebuilding a medical record number in this fashion can be done with a Data Masker Row-Internal Rule. This rule type operates on the entire row of the table and can take information from other columns in the same row and use it to generate a value for a new row. The image below shows a Row-Internal Rule which takes the first three letters of the newly masked *PatientSurname* column and replaces the first three characters of the existing medical record number with the first three characters of the new surname. The last seven digits of the medical record number are generated sequentially from a known start point. This rebuilds the medical record number, preserves the format and also ensures it is unique on the system. Of course, the fact that this rule uses the masked surname value does imply that the rule has to be configured to run after the rule which masks the *PatientSurname* column fully completes and this is simple to do.

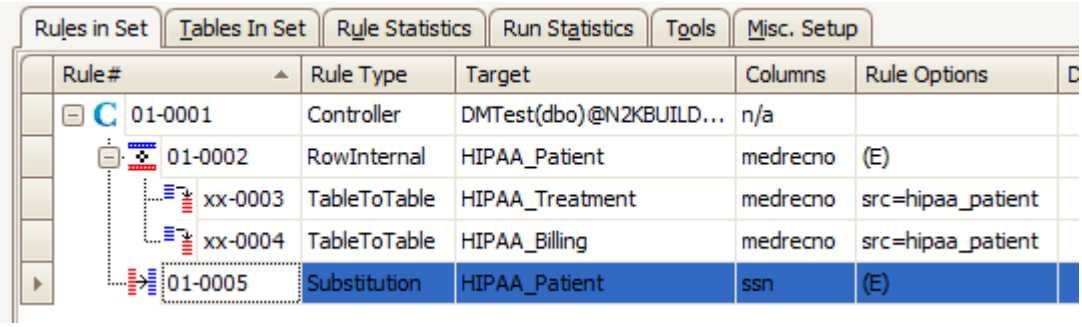


Additional complications arise from the fact that the newly masked medical record number may well be used in many other locations in the database. As with the Substitution Rule illustrated in the *Social Security Numbers* section, it is possible to integrate a Row-Internal Rule into a Sync. Manager Rule and have the new rule fan

out its changes to other locations. It is also possible to use dedicated Table-to-Table Rules to instrument the synchronization in other tables.

The image below shows the Row-Internal Rule with chained Table-to-Table rules. When run, this series of rules will not only rebuild the medical record number but the equivalent medical record numbers in two other tables will also receive the appropriate synchronised values.

It is possible to use the Table-to-Table rules to synchronise changes in any one table into any number of other tables.

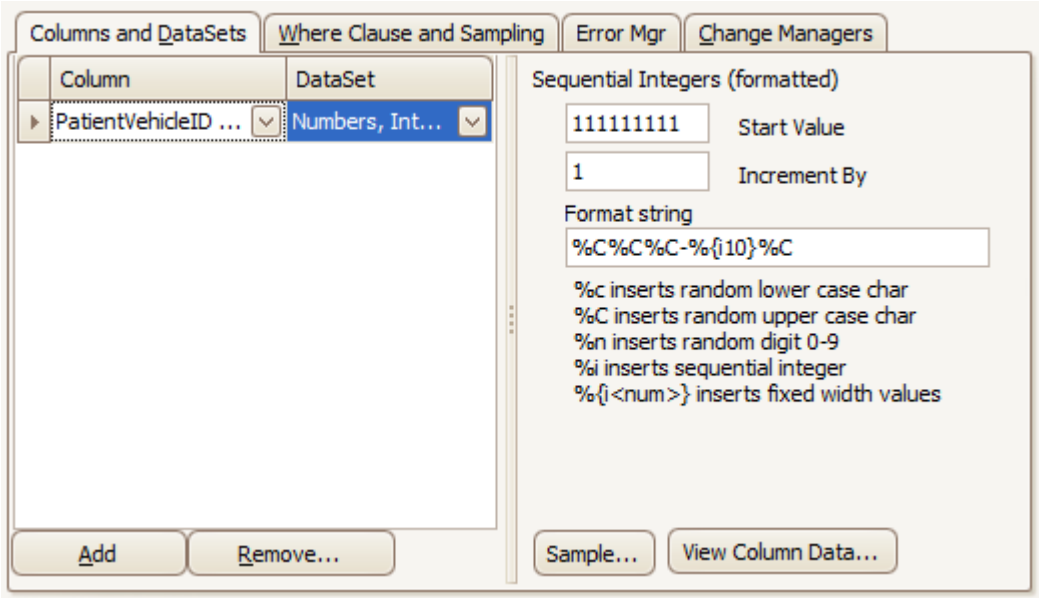


Vehicle Identifiers, Serial Numbers, License Plate Numbers

Identification codes such as vehicle serial numbers or license plate numbers tend to be constructed of collections of numbers and characters and the permitted formats are highly specific. In the case of licence plate numbers each state has its own format and some states also encode county information or month of expiration. Each manufacturer of motor vehicles will have its own format and serial number length.

In some cases, such as motor vehicle serial numbers, where the number is essentially meaningless to the end user of the system it is possible to simply replace all values with randomly generated values and disregard any specific format considerations.

In cases where a number must be regenerated using a specific character, digit and spacing format the Data Masker Text, Formatted or Numbers, Integer, Sequential Formatted datasets can be used. These datasets can output characters and numbers in any specific required format. The image below shows a Data Masker Substitution Rule configured to generate vehicle serial numbers in the format *ABC-1234567890A*



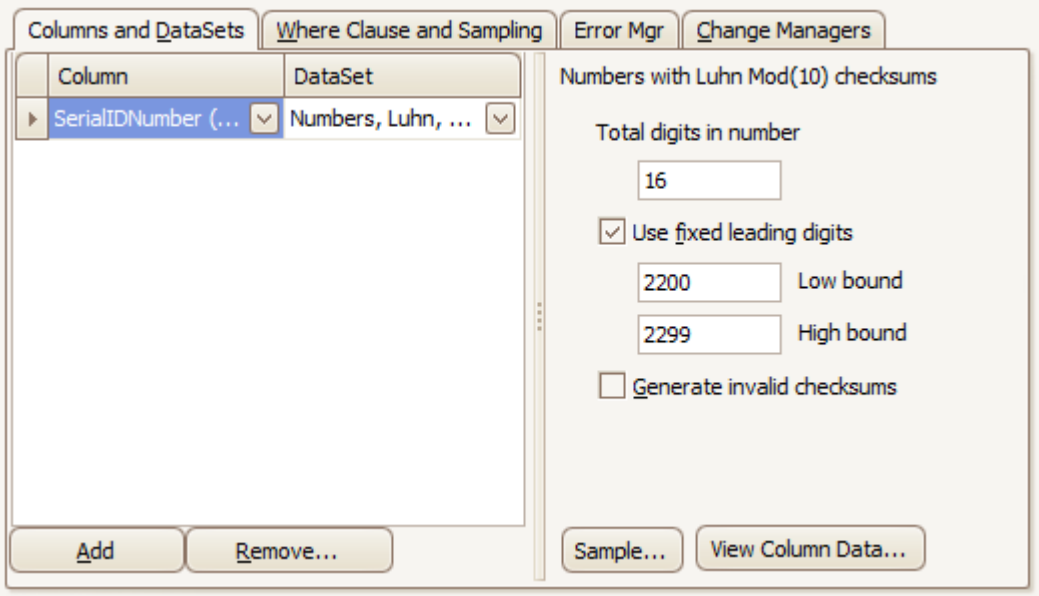
Device Identifiers and Serial Numbers

Device, card and other serial identification numbers of this type often have check digits incorporated within them in order to ensure they are rejected if improperly set during data entry. Credit card numbers are the classic example of this. Most manufacturers use a Luhn based checksum but, in general, each card issuer uses a slightly different variation.

In such cases it is necessary to reproduce a correct checksum if the existing information is replaced with masked data. If this is not done, one runs the risk of creating problems for the end users of the masked system. For example, on some systems, if a credit card number is masked with an invalid checksum and the user attempts to open the customer record to change some other item (such as a telephone number) the front end screen will also attempt to validate the credit card number and will reject all changes until that value is corrected.

The Data Masker software provides datasets which can generate checksum correct credit card numbers for all the major issuers. Configuring such a rule is easy and will not be illustrated here although it is interesting to note that these datasets also contain options to generate guaranteed invalid numbers which are checksum correct. It is also possible to generate unique numbers in such cases where the credit card number is involved in primary or unique keys.

By far the most common type of checksum in use is the Luhn mod(10) checksum and the Data Masker software contains a dataset to generate checksum correct numbers of an arbitrary length. The image below shows a Data Masker Substitution Rule configured to generate a 16 digit number with a Luhn mod(10) checksum. If required, a subsequent Row-Internal Rule could be applied to split it into four groups of four digits with space character separators.



Web Universal Resource Locators (URLs)

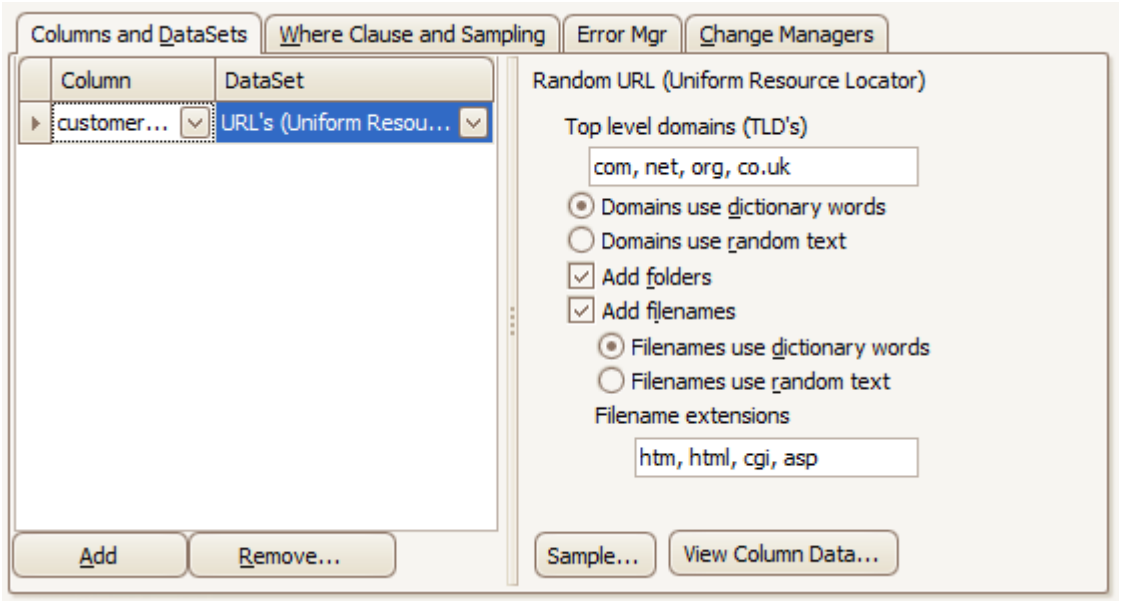
As with most masked data it is usually best to make the resulting masked output resemble the original data and in most situations this will apply to URL data as well. Other than replacing all URL's with some known constant value (or nulls) the remaining options are to manufacture random URLs in various formats or mask them out by replacing some or all of the characters with other characters (such as an 'X').

Replacing a column URL data with other random URLs is pretty straightforward and the only considerations are the type of text data to generate and the structure of the URLs values themselves.

In regards to the type of text, it must be noted that the Internet has been operational for a number of years and pretty much every dictionary word has now been registered as a domain name. If random words are used as domain names, most replacement values are almost certainly going to be pointing at an existing website. The alternative is to use random strings such as "*asdwweh.com*". Whether using real words matters or not in a particular masking situation is a decision which needs to be made at implementation time. Many people feel that it is not a relevant consideration - after all even collections of random characters may well point at an existing domain, particularly if they are short.

URL's can have a variety of formats and in most systems they will probably be stored in a wide variety of formats. For example, URLs could take any of the formats below and there are many other possible formats not shown:

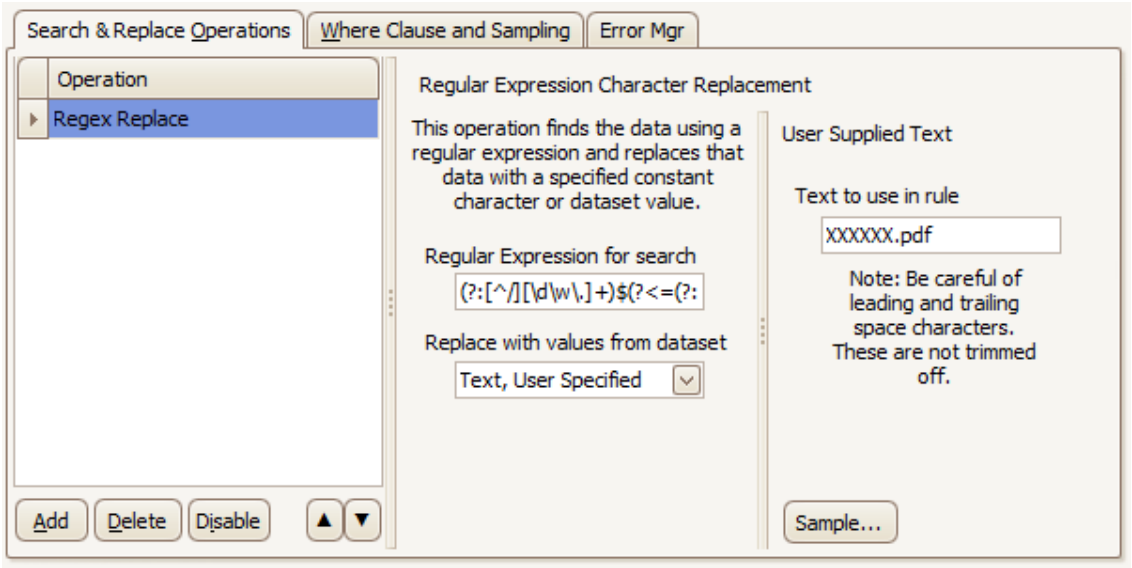
http://DataMasker.com	Top Level Domain (TLD) only
http://www.DataMasker.com	TLD with leading domain(s)
http://www.DataMasker.com/	Trailing '/' on URL
http://www.DataMasker.com/HelpFiles/	Trailing folder but not file
http://www.DataMasker.com/index.html	Trailing filename



Replicating these formats is pretty easy with the Data Masker software. The image above shows a configuration panel for the Data Masker URL dataset. Any of the above formats (or a mixture) can be configured as replacement data and the domain names and the filenames can be dictionary words or randomly generated character text.

Sometimes it is necessary to get quite creative when masking URLs. As an illustration, let's assume there was a requirement to preserve the domain name and folders but mask the filename within the URL itself. For example, the URL *http://www.DataMasker.com/SubFolder/SensitiveDocument.pdf* should be converted to a URL like *http://www.DataMasker.com/SubFolder/XXXXXX.pdf* in which the document names are replaced by the text "*XXXXXX*" but the remainder of the URL is preserved. Operations of this type can be achieved through the use of a Data Masker Search and Replace Rule configured with a Regex Replace Operation.

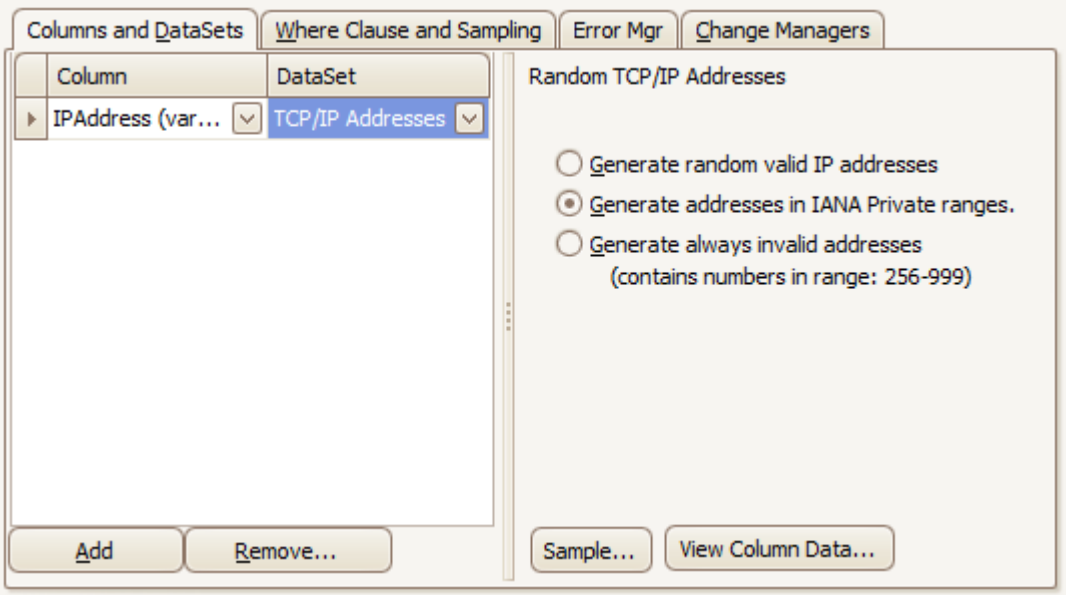
We recognise that writing Regular Expressions for pattern matching is not the sort of thing many people find easy to do – please be aware that the Data Masker support team is always on hand to provide assistance. The image below uses the RegEx pattern `(?:[^/][\d\w\.]+)$(?<=(?:.pdf))` along with the Text, User Specified dataset to mask just the filename in a URL.



Internet Protocol (IP) Addresses

Version 4 IP addresses have long since been consumed and there are very few ranges remaining unassigned. Since IP addresses (unlike URLs) are usually meaningless to the end users of the masked system, there is little benefit in replacing them with randomly generated values since those values could conceivably target a live system somewhere on the Internet. The usual method is to replace IP addresses with a constant known value, values from the IANA Reserved Private Network Ranges or with an invalid value that contains a number greater than 256. The last option, that of forcing the IP address to be invalid by using an impossible number, is only useful in situations where it is known that the resulting masked IP address will never have to pass internal validation checks.

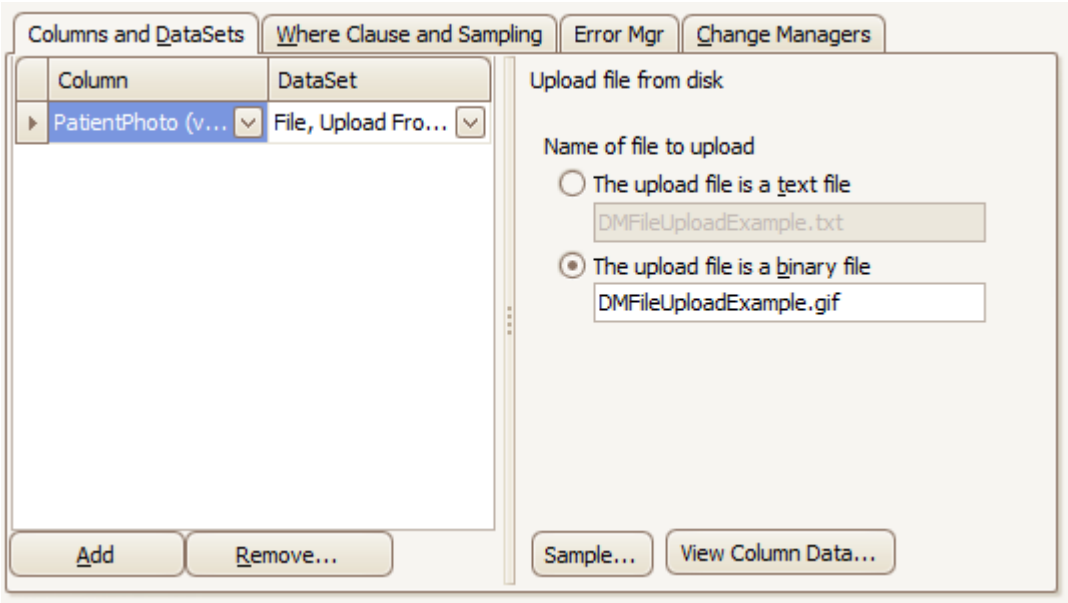
All of the above options are available in the Data Masker software. The image below shows the IP address dataset configured to output random IP addresses randomly distributed among the three IANA Private IP Address Classes.



Full-Face Photographs and any Comparable Images

Most data of this type is stored as a binary object. There really is no practical way to reach inside the image and modify it in such a way as to render it anonymous. The usual practice, besides deleting them entirely, is to replace them with one or more sample images known to be benign. This method ensures that the resulting test system has an image available for viewing in situations where one would be available in the original data but that the image on view is not of a sensitive or personally identifiable nature.

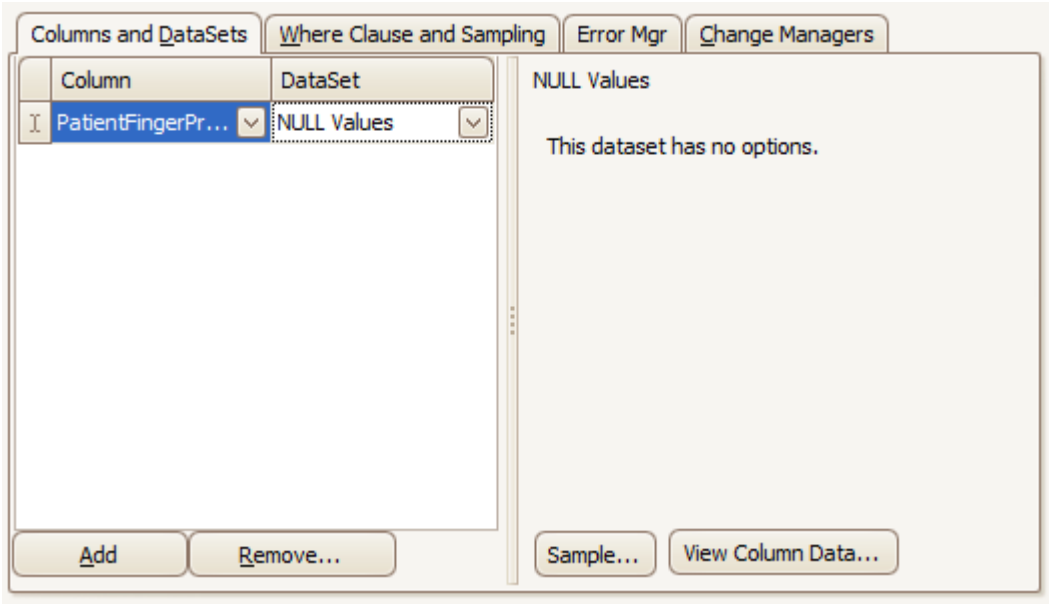
The image below shows a view of a Data Masker Substitution Rule configured to use the File, Upload from Disk dataset to use an example image located on the PC as a replacement for all not null images stored in a remote database table.



Biometric identifiers, Including Finger and Voice Prints

As with the *Full-face Photographs Safe Harbor Method* the usual technique is to replace the existing finger or voice data items with a known anonymous sample file. The implementation technique for that requirement is identical to that used in the *Full-face Photographs Safe Harbor Method* section and is illustrated there. It will not be discussed again here – please refer to that topic for more information.

For the purposes of illustration, if one just wanted to just remove the voice or fingerprint data it could simply be done using the Null Values dataset as shown in the image below.



Other Unique Identifying Number, Characteristic, or Code

This requirement is a typical “catch all” trailer which basically says “Personally Identifiable Information” (PII data) must be masked. The format or content is undefined.

The resolution of this issue breaks down into two parts: the identification of the sensitive PII data and the subsequent rendering of it anonymous. Whilst the identification of the sensitive information is necessarily implementation specific, having versatile masking software available will prove invaluable when the time comes to render the data anonymous. We believe the Data Masker software contains the tools necessary to meet most situations and has been proven in use by hundreds of customers across the world. We are always willing to offer advice and assistance, and in many cases will quickly add functionality and additional data replacement sets to the product in order to address a requirement we had not previously encountered.

