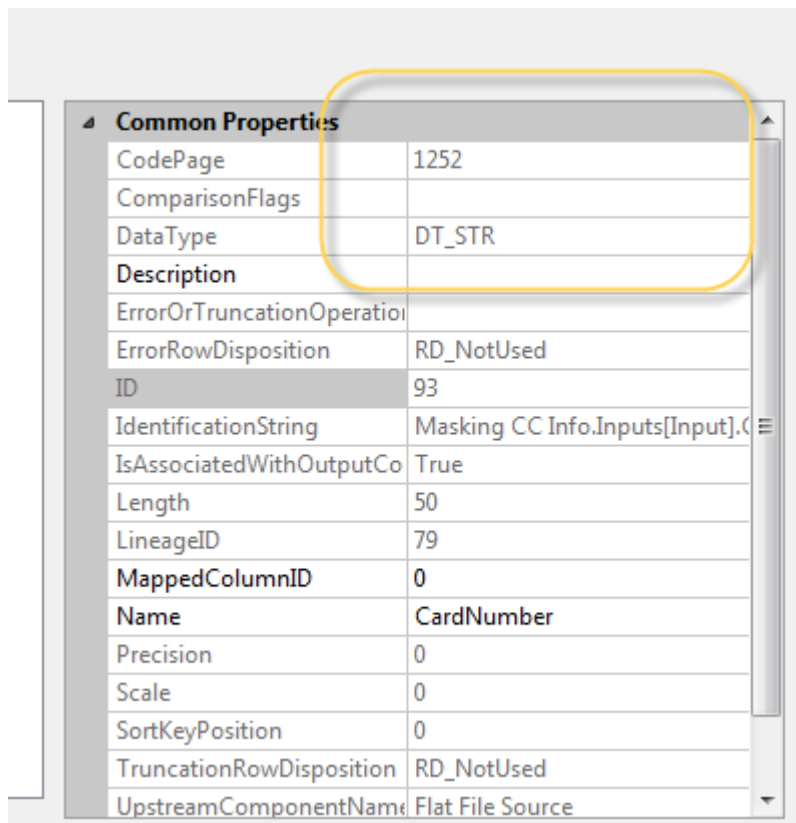


HUSH HUSH MASKING SSIS COMPONENTS MANUAL

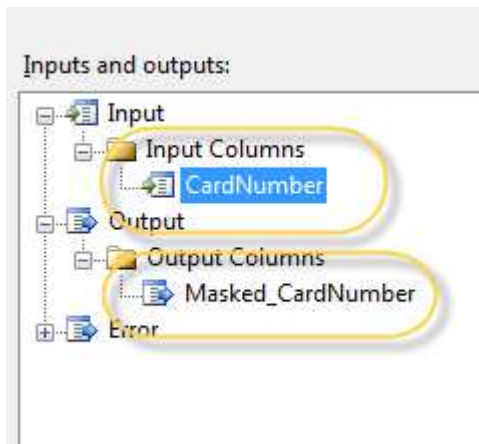
Hush Hush is a collection of SSIS data flow transformation components that encapsulate entities and their rules of generation, with the exception being Generic Alpha Numeric component.

The components expect **string type** data on input and provide **string type** data on output.

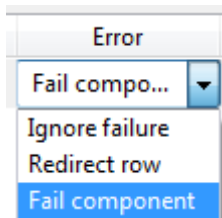


Common Properties	
CodePage	1252
ComparisonFlags	
DataType	DT_STR
Description	
ErrorOrTruncationOperation	
ErrorRowDisposition	RD_NotUsed
ID	93
IdentificationString	Masking CC Info.Inputs[Input].C
IsAssociatedWithOutputCo	True
Length	50
LineageID	79
MappedColumnID	0
Name	CardNumber
Precision	0
Scale	0
SortKeyPosition	0
TruncationRowDisposition	RD_NotUsed
UpstreamComponentName	Flat File Source

Every component allows for one field only on input. It creates another field that has the masked value in it. The name of the new column is a concatenation of the word “Masked”, underscore and input field’s name.



Each component is strongly typed. It expects specific types of data. Upon input, it will confirm data type and rules of formation. Without this information, the component will fail. Three types error handling allow for “fail component”, “redirect row”, and “ignore failure”.



Components employ two major algorithms. One allows for the random substitution of values. The other utilizes proprietary encryption-like algorithms. Both components use entity type in a title. The later ones have the word “Dynamic” in the title.

Random components replace data in a truly random fashion. Values don't get bound to original values; they can also repeat. The randomness introduces the danger of inconsistency among the masked data comparing to original. In order to maintain referential integrity and -when necessary -uniqueness as it is often a case with SSN or credit card numbers, one has to create additional structures, so-called mapping tables. One has to insure that corresponding values are mapped into the table without duplication. There are several ways to achieve this, methods include using distinct clauses at the source, the use of unique indexes and simply by looking up components of SSIS.

Dynamic components retain referential integrity without any additional structures. However, by the nature of encryption, there is a possibility of decryption by unauthorized parties. Therefore, the decision is up to the practitioner on whether to use the “Dynamic” or the “Random” component. Some of the components (with a lesser security threat) at this time only provide the encrypted algorithm. Newer versions will have random algorithms for entities as well.

Only Social Security Numbers, Credit Card numbers, US Phone numbers, and Phone Numbers guarantee that unique data sets will map to a corresponding, unique data set. Names, Address elements, components, URLs, Dates of Birth and Zip Codes do not guarantee unique data set on output but provide sufficient amounts of output values to test upon for various KPIs.

ERROR DISPOSITION

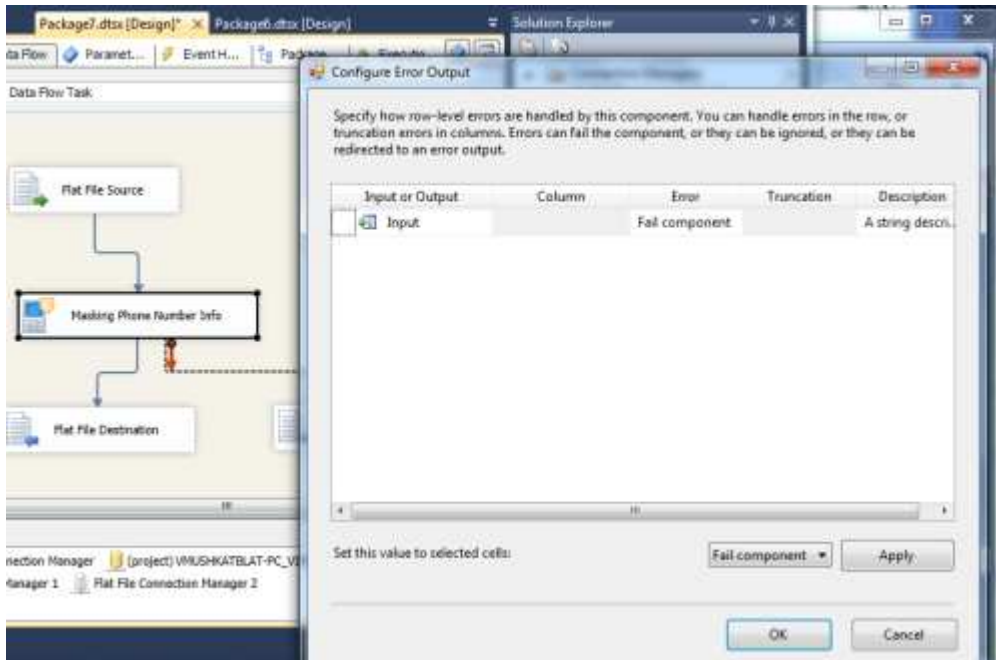
Error disposition works in all the components. It allows you to fail components (by default), to re-direct rows, and to ignore failures. In the first case, the failed component stops the flow. In the case of re-directing, the component directs erroneous rows into the destination that one created to store erroneous content. Ignoring failure will only transfer successfully validated values, without attending to the errors. The “Ignore failure” option might result into the loss of data.

Error disposition rules are limited to the rules of the entity:

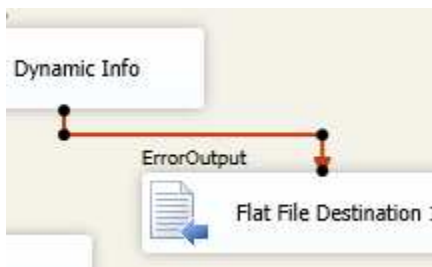
- checks whether numeric components’ (SSNs, Credit Cards, Phones and Zips) input values consist of digits;
- checks that email has a format of [xxx@xxx.xxx](#);
- checks (in a very limited fashion) URL format;
- checks Luhn calculation in Credit Cards.

It is important to notice that all the numeric components assume that non-alphanumeric characters are likely the separators and preserves them intact.

To set error disposition, connect your component with the following component in the flow with the precedence constraint:



The Configure Error Output window opens and contains the “Error” Column. That should be set to the desired behavior. By default SSIS architecture, it does not allow later to open the window again to change set error handling behavior. In order to change it, one has to delete and re-create the precedence constraint.

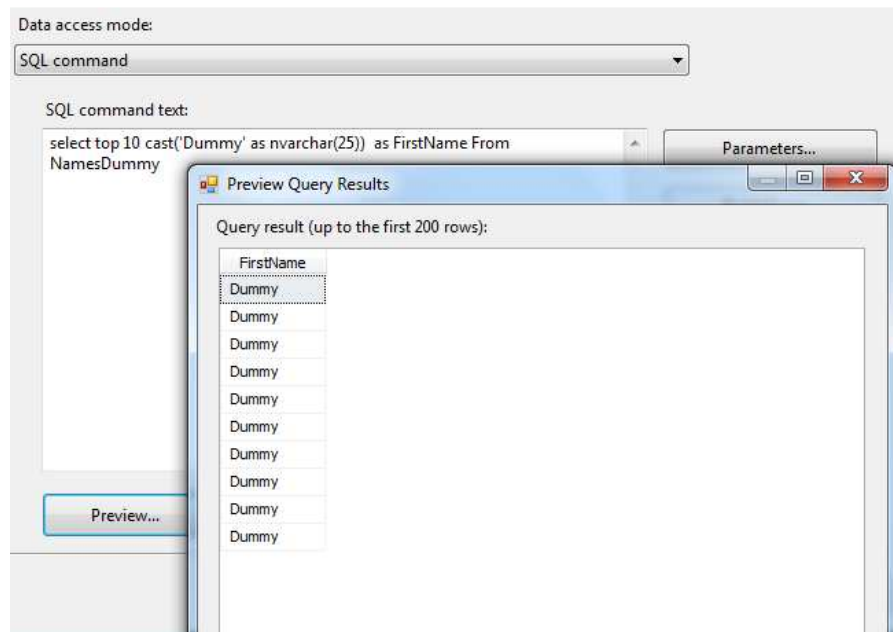


CREATING TEST SETS FROM SCRATCH

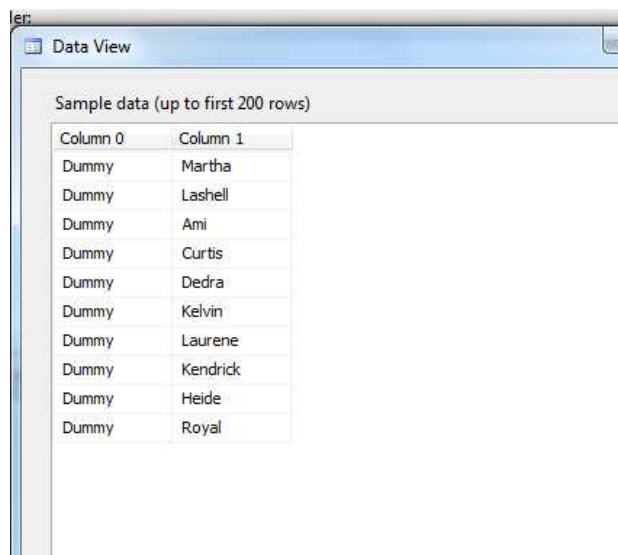
Random components allow you to create test sets when the data in production does not exist. In text-based components such as names and addresses, one simply can provide a “dummy” value in the necessary fields, such as “John” or “abc” – and the component will create a random set of values.

Random components allow one to create test sets when one has created fields in production and does not have data yet. In text-based components such as names and addresses, simply provide a “dummy” value in the necessary fields, such as “John” or “abc” – and the component will create a random set of values. Important note: if one uses SQL query as an input, one has to make sure value that is strongly typed to avoid SSIS conversion issues. See an example:

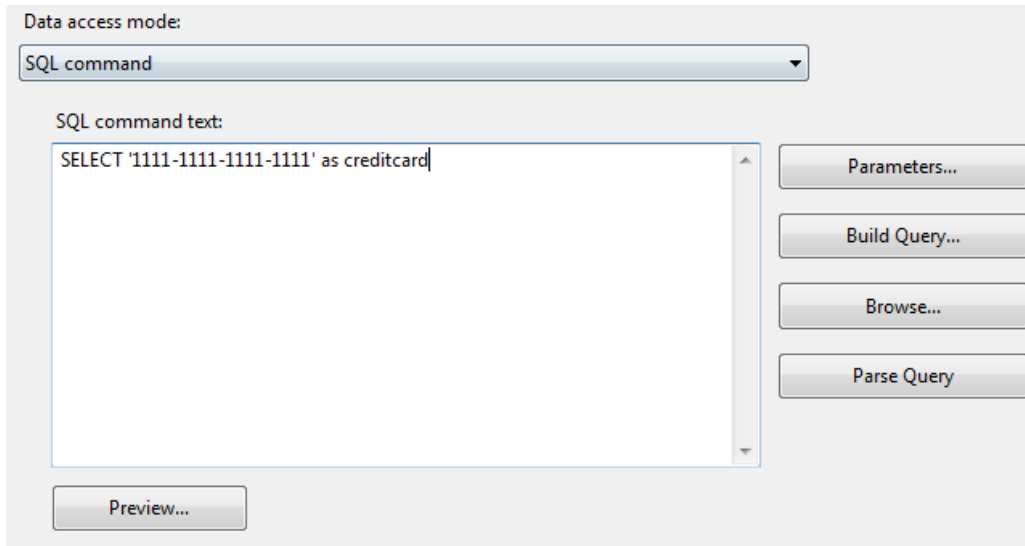
```
SELECT TOP 10 CAST('Dummy' as NVARCHAR(25)) AS FirstName FROM NamesDummy
```



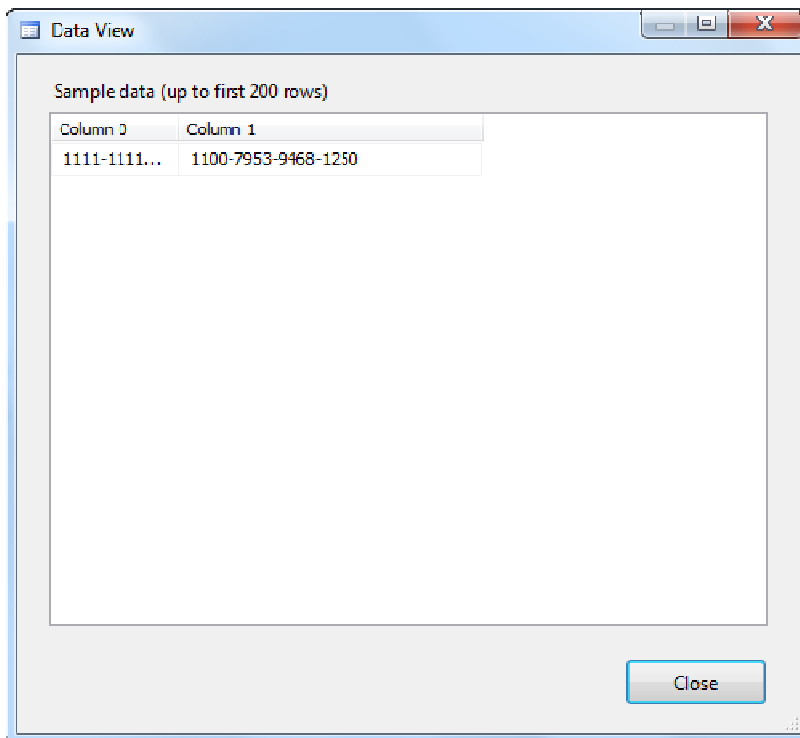
After execution, one will have a list of randomly generated names.



Before creating data for numeric components that require a predefined format (such as credit card numbers) you must create a format that defines the test set:



The records resulting for the example above will preserve the separators and the format. It will only substitute values and validate rules provided by the component. Credit Card, in the above example, maintains the originating agency number and Luhn number, while its length is defined by the practitioner.

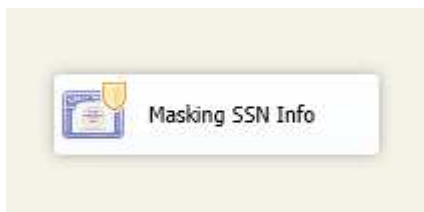


Using SQLstatement to create a “dummy” value will work either way, with cast or no cast and will repeat the number of characters in the dummy.

COMPONENTS FUNCTIONALITY DESCRIPTION

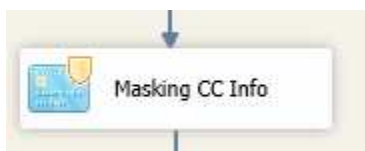


Generic Alpha Numeric components replace letters with letters, maintain capitalization, and replace numbers with numbers. It retains the non-alphanumeric characters, such as dashes, slashes, dots and commas, and keeps track of their position in the string. It currently only replaces characters from alphabets that uses English characters. It will replace strings based on letters only with letter strings and strings based on numeric characters, only with numeric characters.



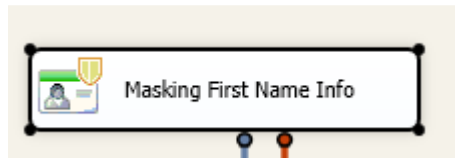
The SSN component creates a social security number. It generates only real social security numbers. There will be no numbers that start with 000,666, 900-999. It expects input in AAA-AA-AAAA format.

The SSN Dynamic component maintains all the features of the SSN component and can be used without creating a masking table. The component replaces values on the fly. The SSN Dynamic component recognizes that your current system indeed might have testing records starting with 000,666, 900-999 prefixes. If it encounters these prefixes, it keeps the prefix intact, masking the rest of the number.



The Masking Credit Card component expects a credit card number in a string that consists of integers, ie: XXXXXXXXXXXXXXXX. There might be separators in between the digits, such as “-“ or “/” . It produces a card of the same issuer with the parity digit consistent with Luhn calculation.

The Masking Dynamic Credit Card component (“Masking CC Dynamic Info”) is transforming credit card numbers consistently using FPE –like algorithms, while retaining the issuer and Luhn.



The Masking First Name component expects First Name as an input and produces First Name as an output. The First Name Dynamic input will produce consistent First Names, if the name repeats, the output will repeat the same name in accordance with original mapping. It does not provide uniqueness.



The Masking Last Name component expects Last Name as an input and produces Last Name as an output. The Last Name Dynamic input will produce consistent Last Names, if the name repeats, the output will repeat the same name in accordance with original mapping. It does not provide uniqueness.



The Masking Address component takes street address as an input and produces a street address as an output in form of <Street Number> <Street Name> <Street Predicate>. There is only a Dynamic component in this release. It does not guarantee uniqueness but uses more than a million combinations in creating addresses.



The Masking City Names component changes The City name into another random city name chosen from a list that excludes the names of America's 1000 largest cities. The component guarantees consistent transformation of values but does not guarantee uniqueness.



The Masking Zip Code component consistently changes zip codes to the values of the zip codes that are currently used. Unused zip codes on input will result into failure. The component expects a 5 digit string. Alpha characters on input will result into failure. In future releases there will be a choice between five digit zip code and seven and nine digit zip codes and postal codes of other countries. Currently, if you want to mask a zip code containing alphanumeric combination, just use Generic Alphanumeric component.



The Masking Date of Birth component provides generic date handling. It expects a string in the date format ('MM/DD/YYYY) and can be used to mask any date. However, one has to be aware that per HIPAA requirements dates of birth that exceed 89 years age limit are going to output a date that does not allow discovery of age. The component also allows selection of the interval of confidence in days that defines interval of variation for the date. The date itself is in the middle, interval values add both positive and negative numbers of days to the date. Example: if May, 1st 2000 is the date, defining an interval of confidence as two days will produce April 29th, April 30th, May 1st, May 2nd, and May 3rd as values.

Component Properties | Input Columns | Input and Output Properties

Configure advanced properties for the data flow component.

Properties:

Common Properties	
ComponentClassID	{874F7595-FB5F-40FF-96AF-FBFF8250E3EF}
ContactInfo	Masking Service Company
Description	A SSIS Data Flow Transformation Component To Provide Date Of Birth
ID	95
IdentificationString	Masking Date Of Birth Info
IsDefaultLocale	True
LocaleID	English (United States)
Name	=
PipelineVersion	0
UsesDispositions	False
ValidateExternalMetadata	True
Version	0
Custom Properties	
Confidence interval (in days)	30
UserComponentTypeName	SSIS.Components.MaskingDateOfBirthInfo.MaskingDateOfBirthInfo, M



The Masking US Phone Number component expects the data to be in the format of xxx-xxx-xxxx where x is an integer. The output value retains the area code and masks the phone number. Currently there is only the “Dynamic” algorithm and this algorithm allows you to retain consistency and uniqueness of your values upon output.



The Masking Phone Number component is a Dynamic component that expects valid international phone number starting with the international country code. It will process numbers both starting with plus (+) and not. The component allows a variety of separators as formatting including “-“,”,.” and spaces. The country code prefix will remain unchanged.



The Masking Email component expects data in [xxxx@xxxx.xxx](#) format. It provides output in the same format, with .com domain. It provides consistency in the output, meaning given the same input, it will provide the same output. However, collisions are possible although highly unlikely. While it is not difficult to create unique emails using names and sequence numbers with the derived component's string functions, the Email component provides a quick and dirty way to create consistent masked emails.



The Masking URL component expects a URL-like string upon input and provides a value for the URL upon output. The Dynamic component provides a choice for financial institutions URLs, health institutions URLs, or mix. To chose the type of URLs, one has to put 1,2, or 0 correspondingly in the custom property of company type field in the editor:

Version	0
Custom Properties	
Company Type	0
UserComponentTypeName	SSIS.Components.Mas

By default “Company Type” = 0 and presents a mix of URLs.

INTRODUCED IN THIS RELEASE:

- Credit Card Dynamic Component
- International Phone Number Component
- Error disposition feature is introduced in all the components
- Nulls are preserved (reported bug, fixed)

KNOWN ISSUES:

Zip component expects only used zip codes on input, will be fixed to allow to re-direct the rows.
Currently redirect only consistently works with files sources.

COMING RELEASE NEW FEATURES:

Random Components will be introduced for Phone Number, Cities, Addresses and Zip Codes.

Expiry Date components will be introduced that allows date data type on input.

Zip Code component will allow seven and nine digits.

More components for masking company names, passports and bank account numbers.

CUSTOM COMPONENTS

Please contact us if there is a custom component that you need **in your particular situation!**

TABLE OF COMPONENTS AND ALGORITHMS

COMPONENT	RANDOM	DYNAMIC
SSN	X	X
Credit Card	X	X
First Name	X	X
Last Name	X	X
Address		X
City Names		X
US Phone Number		X
Email	X	
ZIP		X
Date of Birth	X	
URL	X	X
Generic Alpha Numeric		X
Phone Number		X