Test Data Management

A Whitepaper by
Gopinath Murali
DBA Group Manager, Mphasis Data Engineering Practice

Testing industry is always looking for ways to optimize the efforts and costs they invest on testing,and this leads to arise of Test Data Management (TDM) which provides integrated sensitive data discovery, business classification and policy-driven data masking for de-identification and safe use of production data used in test and development environments.

This Whitepaper addresses the significance of testing and masking, need for test data and data masking, and how it is performed.

Abstract

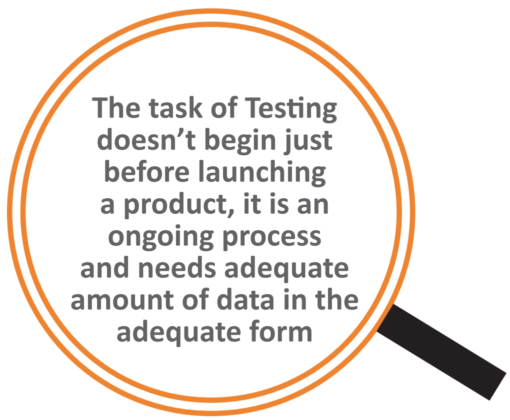
The purpose of this paper is to discuss on the methods and means by which the data can be generated and masked to make it available for test. Today, testers need an in-depth knowledge on IT architectures and testing methodologies. A sample mobile app illustrates the phenomenon. The app enables the customers to do intra-day trading. It send trades to bank core system, which then forwards them to stock exchange. End-to-end testing becomes a real challenge. Testing must consider the mobile platform and the smartphone type before testing the interface and then finally test the test data on the app, the core banking system and the exchange server.

Recent survey clearly states that 60% of application development and testing time is devoted to data related tasks, making it cumbersome and time consuming. For organizations to speed-up testing and control costs, they need well captured test data requirements and the right test data management strategy. Without these in place, the entire process of following the exact methodologies of test planning, scripting and automation will be lost. By using the right TDM strategy QA organizations are able to provide a consistent and standardized approach to testing applications.

The significance of Testing and Masking

One of the biggest challenges in building enterprise application is to deliver a robust, stable and bug-free application. Although there are many key factors that help us achieve this, the substantial factor is testing. Testing teams not only have to follow exact test methodologies, but also ensure accuracy of test data. They also need to ensure that these tests correctly reflect production situations both functionally and technically.

Whether you're testing new features and functions in a system, or validating defects reported by a client, it is vital that you can accurately emulate the workflows and interactions as encountered by the end-users. To do that, you should be able to synchronize the use of correct and accurate data during the tests. You can increase the faith in your testing (thereby also increasing the quality of your test coverage) by getting the correct data in the right quantity, at the right time within the system. So, how better could it be done to reflect real-world scenarios? Welcome to the world of test data management!



Why Test data is required

Depending on the requirements of the application, test data is either created or duplicated from the production environment. In such scenarios, test data is created in-sync with the test case it is intended to be used for. This sort of test data will not help in identifying the bugs. Since minimal test data is created, performance of the application cannot be tested. Because of these drawbacks, generating the test data similar to production environment becomes a crucial need. When it comes to the financial applications, **"Data Masking"** plays a vital role in test data generation.

Many applications contain sensitive personal information. There may be government mandates and regulations in place that stipulate the data that must be masked, de-identified, or encrypted. Without a solid process to protect that data, there's a real risk that valuable and personal information could leak and can be used in a malicious manner. A data breach can be extremely expensive to sort out; it can damage reputation and result in lawsuits and punitive fines. This happens mostly when the production data is replicated. These kind of scenarios can be handled with the concept of "Data Masking".

Source of Test data

Today IT departments are aware that production databases store sensitive data. Business users see only the data required for their work. In contrast, IT departments often ignore the risk of test servers; most of them contain production data. The data is copied periodically or on request from production servers. As a result, many developers and testers can potentially access production data on test servers. This involves high chances of data loss and Test data management project should mitigate this risk.

Example scenario 1

Consider there is a company XYZ which begins with 1000 users per month and in one year the number of users per month grows to 10,000; the number of users in 3 years will be 3 x 12 x 10000. Now if the existing system is functioning in Oracle and the client want to move the system to Sybase, then generating that huge volume of data in Sybase and performing the test is very important. At the same time, copying the data from the Oracle database to Sybase database itself consumes more time than testing. The data in the test data generation will play a significant part, as the client information is crucial.

To overcome this we have developed a job in **Talend Open Studio** for **Data Integration** to generate test data. Talend is an open source application for data integration job design with a graphical development environment. Since it's an open source tool, it gives more flexibility to extend the attributes of components based on our requirement.

How to generate Test data using talend

Below components are used to generate the test data to handle the business scenario.

- **tRowGenerator:** Generates as many rows and fields as required using random values with the help of the user-defined routine. Below is the code sample of the routine, which can be customized as per the requirement. We can add as many methods as we need to this routine.

Routine

It contains the following methods:

getFirstName – Gets the random first name

getLastName – Gets the random last name

getUsStreet – Gets the random street name

getUsCity – Gets the random city name

getUsState – Gets the random state name

- **tMap:** This component derives the column value and mapsto the target table with respective columns. The transformation is done here.

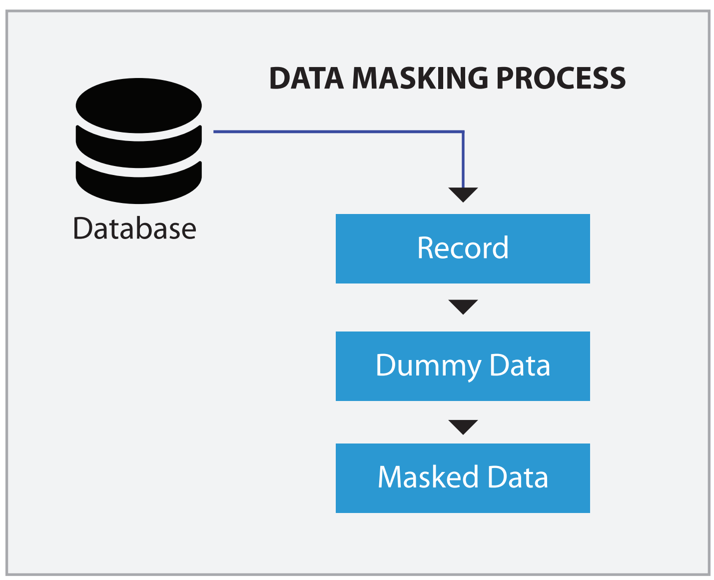
Attributes	Derivations	Sample Data
sno	Numeric sequence ("s1",100000,1)	100001
fname	MyTestData.getFirstName()	Viswanathan Anand
lname	MyTestData.getLastName()	Reagan
ssn	Numeric.random(100,999)+ "-"+Numeric.random(10,99)+ "-"+Numeric.random(1000,9999)	670-73-3525
dob	TalendDate. getRandomDate("2007-01-01", 2008-12-31")	06-AUG-08
sex	MyTestData.getSex()	Male

- **tOracleOutput:** This component gets the connection details from the metadata repository and loads the data to the respective table.

Data masking in talend

In TDM, data masking helps in managing data across development, testing, training and reporting environment. It allows actual data to exist in databases, which remains cloaked before it reaches users without the need for security clearance required to access the proprietary information.

There can be many ways in which data masking can be implemented. It could be a substitution of existing records with expected test data or shuffling of certain characters or numbers, thus generating a new record. Alternatively it could be as complex as using proprietary algorithms to scramble or obfuscate a part of the record with a random data generated using the algorithm, which has all properties that original data had.



Following are the ways using which data masking can be achieved in talend:

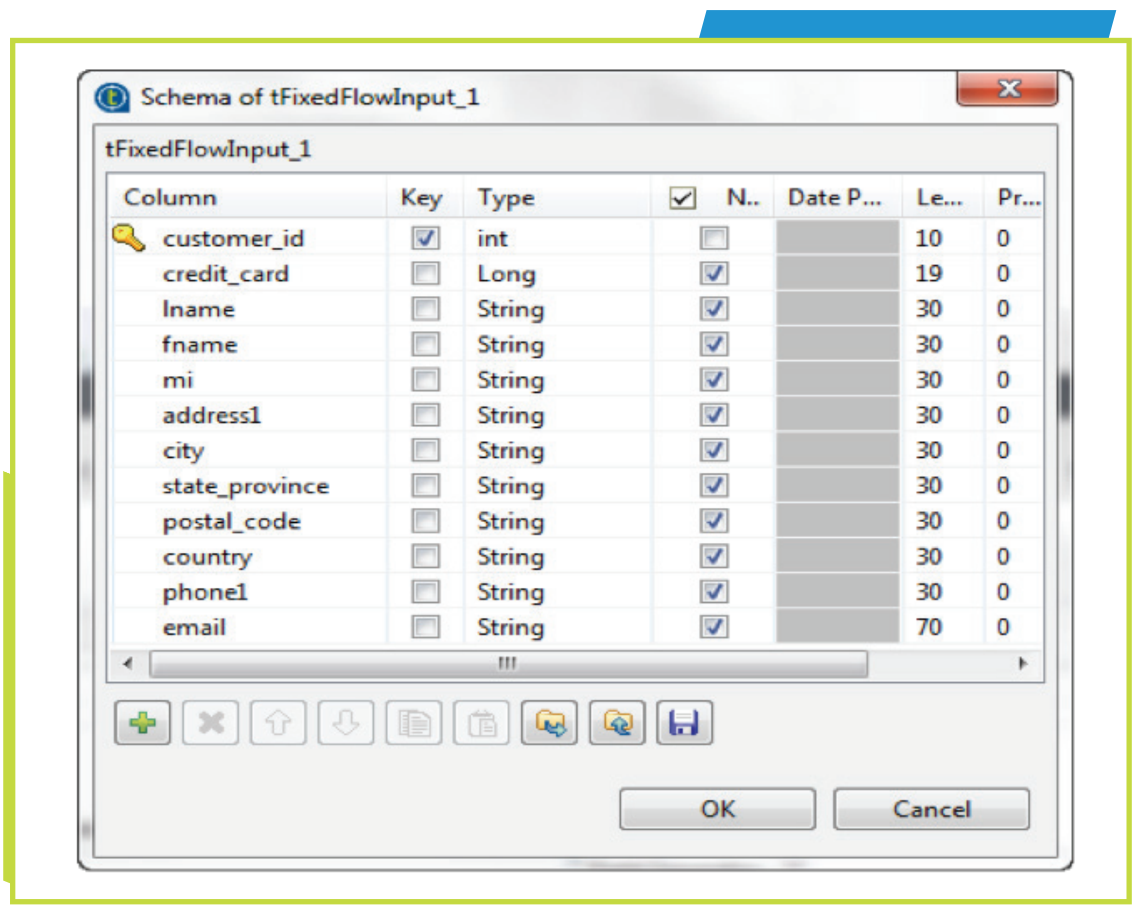
- Using **tDataMasking** component
- Encrypting sensitive information like password
- Transforming the sensitive data using **tMap** component
- Writing Java routines to transform the actual data
- Lookup replacement

Example scenario 2

Consider there is a source file that consists of production data. Testing environment has to use this production data from the source file for testing purpose. In such case, the sensitive data in the source file has to be protected from being exposed while testing.

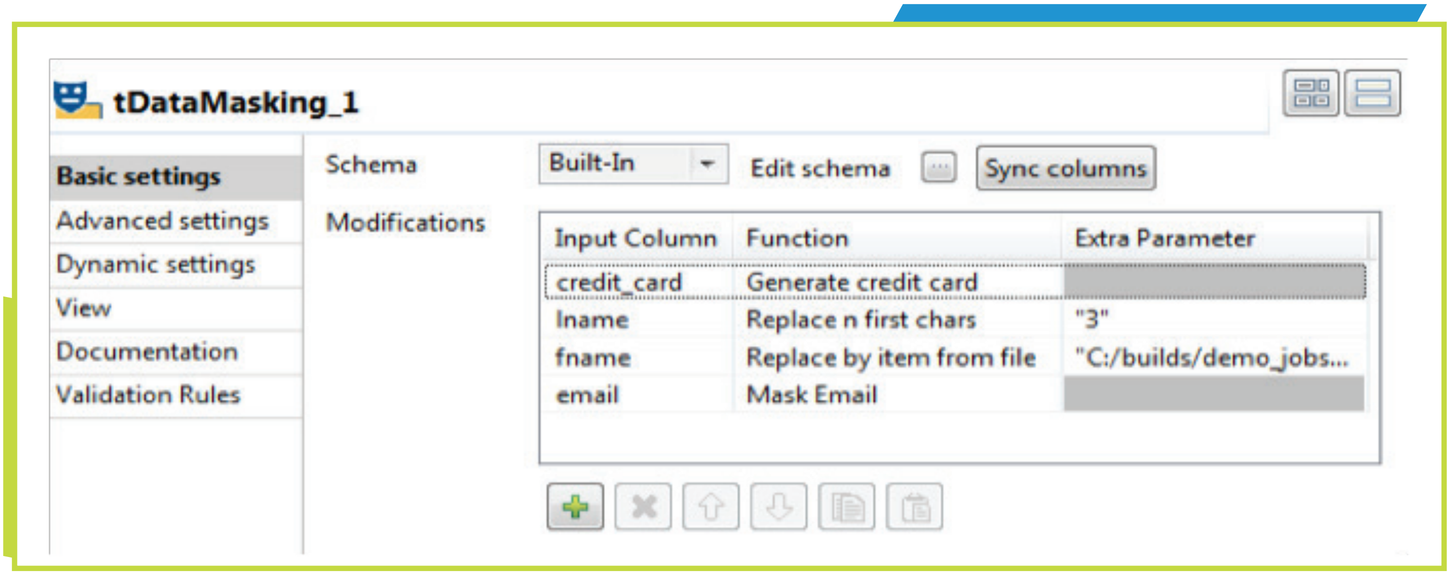
Below components can be used for implementing data masking using Talend -

- **tFixedFlowInput:** The schema of the input file looks similar to the schema below which consists of some sensitive information such as credit card number -



Column	Key	Type	N..	Date P...	Le...	Pr...
customer_id	☑	int	☐		10	0
credit_card	☐	Long	☑		19	0
lname	☐	String	☑		30	0
fname	☐	String	☑		30	0
mi	☐	String	☑		30	0
address1	☐	String	☑		30	0
city	☐	String	☑		30	0
state_province	☐	String	☑		30	0
postal_code	☐	String	☑		30	0
country	☐	String	☑		30	0
phone1	☐	String	☑		30	0
email	☐	String	☑		70	0

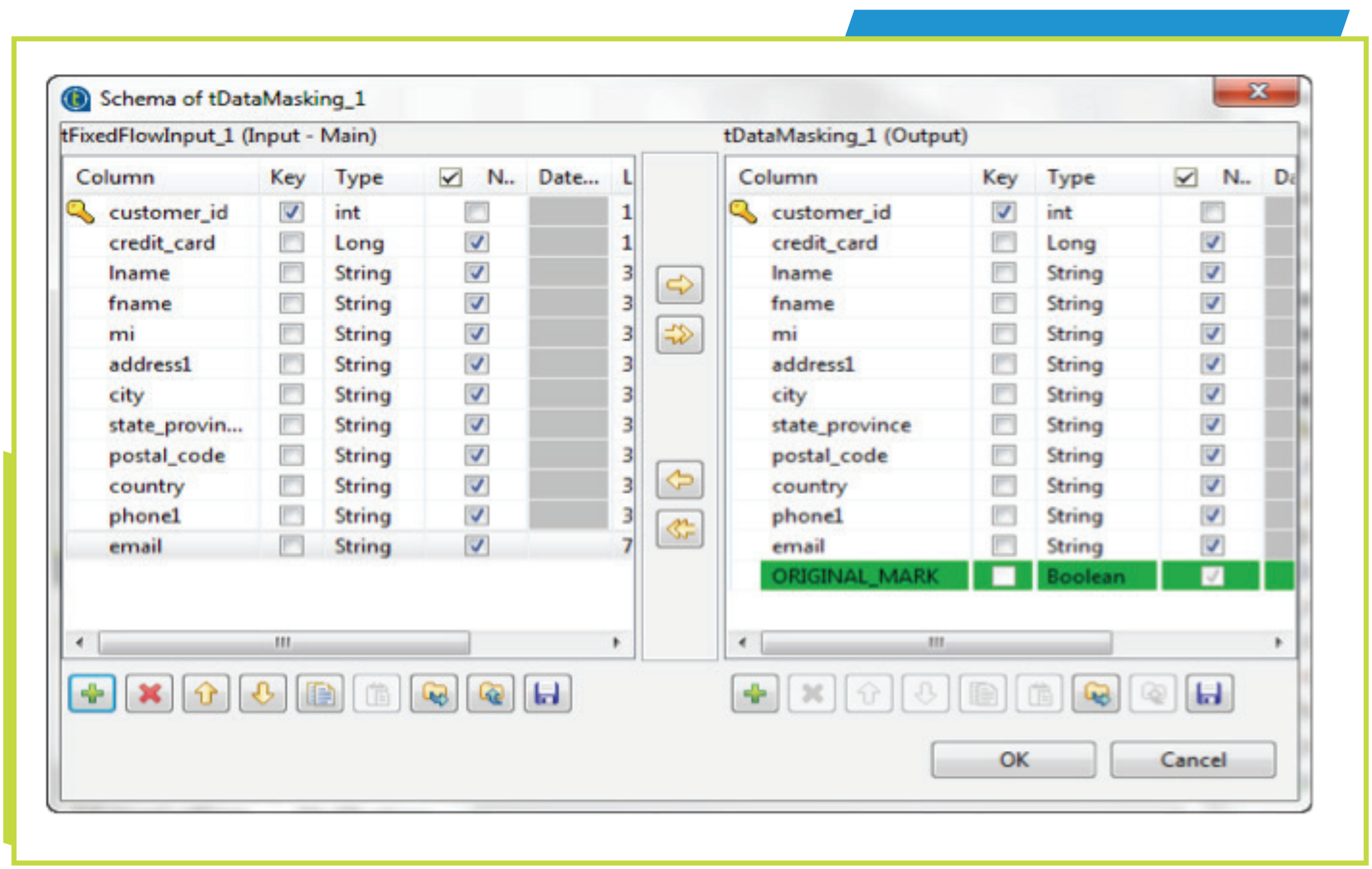
- **tDataMasking:** In data masking component you mask the sensitive data



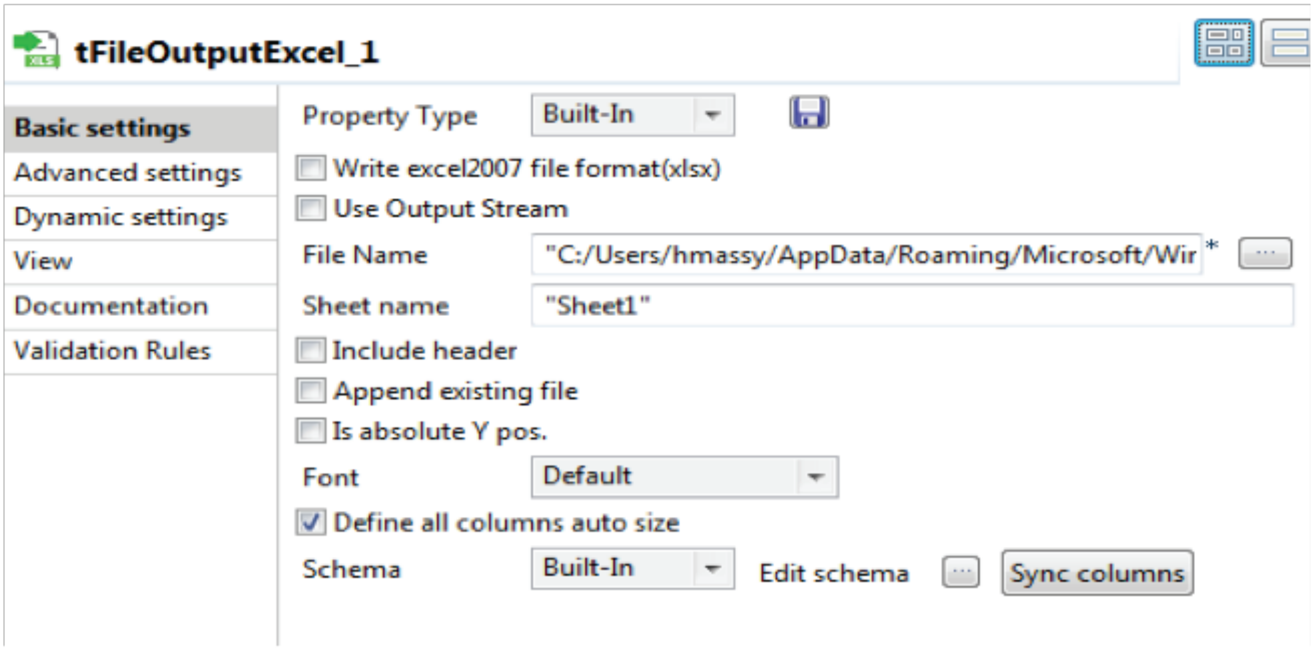
In the **Modification** table, click the [+] button to add four rows, and then:

- In the **Input column,** select the columns whose content you want to substitute
- In the **Function column,** select from the predefined list the function you want to use to generate the substitute data
- In the **Parameter column,** enter a value, a pattern or a path to be used by the function to substitute data

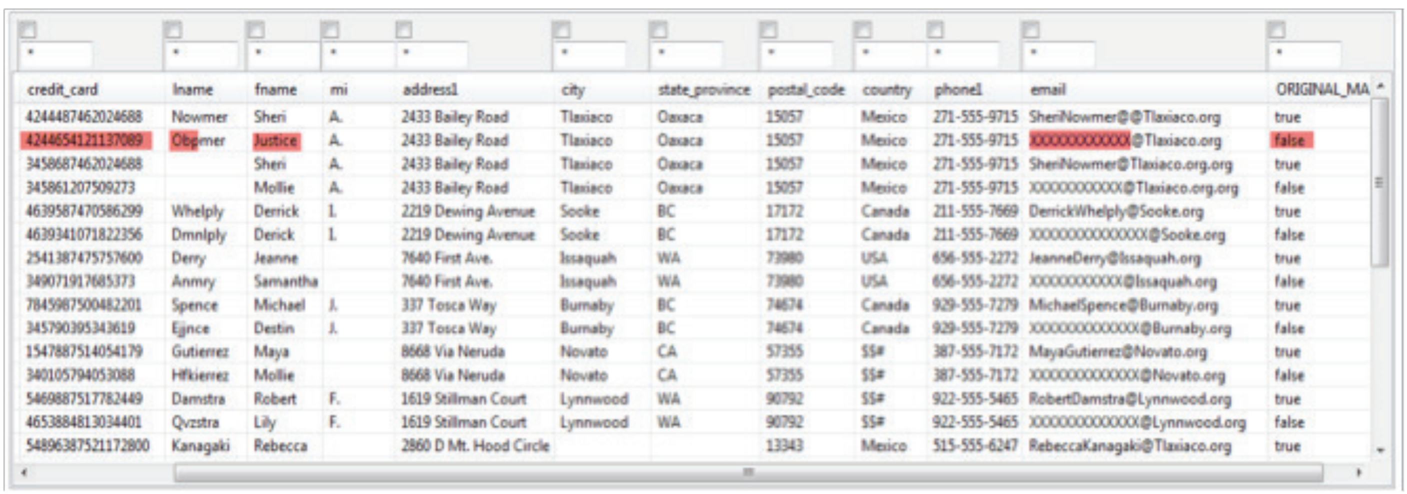
After modifications, the final schema of data masking component looks like this -



- **tFileOutputExcel:** The properties of tFileOutputExcel looks like below -



After defining the tFileOutputExcelcomponent properties, we execute the job and get the output as follows -



credit_card	lname	fname	mi	address1	city	state_province	postal_code	country	phone1	email	ORIGINAL_MA
4244487462024688	Nowmer	Sheri	A.	2433 Bailey Road	Tlaxiaco	Oaxaca	15057	Mexico	271-555-9715	SheriNowmer@@Tlaxiaco.org	true
4244654121137089	Obpmer	Justice	A.	2433 Bailey Road	Tlaxiaco	Oaxaca	15057	Mexico	271-555-9715	XXXXXXXXXXXX@Tlaxiaco.org	false
3458687462024688		Sheri	A.	2433 Bailey Road	Tlaxiaco	Oaxaca	15057	Mexico	271-555-9715	SheriNowmer@Tlaxiaco.org.org	true
345861207509273		Mollie	A.	2433 Bailey Road	Tlaxiaco	Oaxaca	15057	Mexico	271-555-9715	XXXXXXXXXXX@Tlaxiaco.org.org	false
4639587470586299	Whelply	Derrick	I.	2219 Dewing Avenue	Sooke	BC	17172	Canada	211-555-7669	DerrickWhelply@Sooke.org	true
4639341071822356	Dmnlply	Derick	I.	2219 Dewing Avenue	Sooke	BC	17172	Canada	211-555-7669	XXXXXXXXXXXXXX@Sooke.org	false
2541387475757600	Derry	Jeanne		7640 First Ave.	Issaquah	WA	73980	USA	656-555-2272	JeanneDerry@Issaquah.org	true
349071917685373	Anmry	Samantha		7640 First Ave.	Issaquah	WA	73980	USA	656-555-2272	XXXXXXXXXXX@Issaquah.org	false
7845987500482201	Spence	Michael	J.	337 Tosca Way	Burnaby	BC	74674	Canada	929-555-7279	MichaelSpence@Burnaby.org	true
345790395343619	Ejnce	Destin	J.	337 Tosca Way	Burnaby	BC	74674	Canada	929-555-7279	XXXXXXXXXXXXX@Burnaby.org	false
1547887514054179	Gutierrez	Maya		8668 Via Neruda	Novato	CA	57355	$$#	387-555-7172	MayaGutierrez@Novato.org	true
340105794053088	Hfkierrez	Mollie		8668 Via Neruda	Novato	CA	57355	$$#	387-555-7172	XXXXXXXXXXXXX@Novato.org	false
5469887517782449	Damstra	Robert	F.	1619 Stillman Court	Lynnwood	WA	90792	$$#	922-555-5465	RobertDamstra@Lynnwood.org	true
4653884813034401	Qvzstra	Lily	F.	1619 Stillman Court	Lynnwood	WA	90792	$$#	922-555-5465	XXXXXXXXXXXXX@Lynnwood.org	false
54896387521172800	Kanagaki	Rebecca		2860 D Mt. Hood Circle			13343	Mexico	515-555-6247	RebeccaKanagaki@Tlaxiaco.org	true

tDataMasking outputs original and substituted rows marked respectively with **true** and **false** in the **ORIGINAL_MARK** column. It generates inauthentic credit card numbers, replaces the first three letters of last names, replaces first names with names from a local file and finally replaces the part before the @ sign in email addresses by a series of X.

Sensitive personal information in the input data is “hidden” but data always look real and consistent. The substitute data can be always used for all purposes other than production.

Gopinath Murali

DBA Group Manager, Mphasis Data Engineering Practice

Gopinath has more than 17 years of experience in the IT industry, with expertise in administering Oracle databases, building High Availability solutions like Data guard and Real application clusters, near down-time [OR] Zero-downtime upgrade/migrate solutions, Oracle Replication using GoldenGate and Identity management solutions. He is currently the Group Manager and Solution Architect for TDM and Data masking, part of Mphasis Data Engineering Practice. He holds a Master Degree in Computer Applications (MCA) from University of Madras.

About Mphasis

Mphasis (BSE: 526299; NSE: MPHASIS) applies next-generation technology to help enterprises transform businesses globally. Customer centricity is foundational to Mphasis and is reflected in the Mphasis' Front2Back™ Transformation approach. Front2Back™ uses the exponential power of cloud and cognitive to provide hyper-personalized ($C = X2C^{2}_{TM} = 1$) digital experience to clients and their end customers. Mphasis' Service Transformation approach helps 'shrink the core' through the application of digital technologies across legacy environments within an enterprise, enabling businesses to stay ahead in a changing world. Mphasis' core reference architectures and tools, speed and innovation with domain expertise and specialization are key to building strong relationships with marquee clients. To know more, please visit www.mphasis.com

For more information, contact: marketinginfo@mphasis.com

USA
460 Park Avenue South
Suite #1101
New York, NY 10016, USA
Tel.: +1 212 686 6655

UK
88 Wood Street
London EC2V 7RS, UK
Tel.: +44 20 8528 1000

INDIA
Bagmane World Technology Center
Marathahalli Ring Road
DoddanakundhiVillage, Mahadevapura
Bangalore 560 048, India
Tel.: +91 80 3352 5000

www.mphasis.com

VAS 24/02/17 US LETTER BASIL4304