

Data Privacy and Test Data Deployment

Software Development Process

Through legislation and good business practices, the desire for data privacy has expanded to include the software development processes itself. Part of that process involves obtaining and maintaining high-quality test data. The issue of data privacy impacts test data in that the potential exists for unauthorized access to sensitive or private data during the testing phase of software development. Although software engineers are an honorable lot, they don't require access to this sensitive or private information for testing purposes- or do they? What about a simple production system failure? What about system test data that is currently a sampling of the data in production? How can software problems be resolved without the data that caused them to malfunction? How can new and existing systems be tested and maintained without representative test data?

Problem Resolution and Other Software Testing

The only reliable way to solve a software problem is to be able to replicate it in a controlled environment. In an ideal world, software engineers should be able to recreate the problem with the live system using some set of test cases they already have. If not, they could make some educated guesses as to what the problem was and attempt to recreate it through trial and error by creating or using some possible existing test cases and modifications to them. This can be a time consuming and expensive process. To be expedient, software engineers have learned to obtain the data required for the malfunctioning process from the live system and put it in the test environment for problem resolution. Using this live production data, however, violates the tenets of both data privacy legislation and good data privacy practices.

As it turns out, the specifics of who and what are far less important in problem resolution of this kind than simply the data structures and numeric values that caused the problems. It doesn't matter to the application or the software engineer whether it was the president of the company whose salary caused the payroll application to malfunction or whether it was the janitor's. It doesn't matter that invoice 27 died because of an order for nuclear bomb triggers or an order for widgets.

This is true for all kinds of software testing - unit testing, string testing, system testing, acceptance testing, volume and performance testing. Software and software testing doesn't care about the who, what, when and where of the live production values, but rather the size of the data values and the combinations of events. Yet, development continues to use live data for their testing needs, as this is often the easiest data to obtain and it includes the data structures needed. However, data stripped of its meaning to the real world can be just as useful in performing software tests and resolving problems and can also satisfy the requirements of data privacy. Many software development organizations have realized this and have chosen to limit the use of live production data for testing purposes through various techniques of Data Masking.

Data Masking

Data that has been stripped of its real world meaning has many names. Some refer to it as Masked, Jumbled, Encrypted, Scrubbed, De-Identified or even Sanitized. Many of these names come from the processes used to make the data void of its real world meaning, but still useful in software testing. This transformation process has a number of challenges and considerations to take into account:

- **Consistency** - making sure that the same data is transformed in a consistent way so its relationships are maintained. If a customer's name or an address is changed in one place, consistency requires that it be changed in the same manner in all places that refer to that customer. If an employee number is changed in one place, guarantees must enforce that altered value for the same employee in other places.
- **Uniqueness Constraints** - masked data must support uniqueness constraints. If a social security account number used as a unique key is changed, not only does it have to be changed to the same value everywhere it is referenced, but it also has to maintain the uniqueness property with respect to other social security account numbers in the entire system.
- **Referential Integrity** - even well known system assigned values may have to be changed but in a consistent way. Product numbers, Customer numbers, account numbers, are examples. Any data value used as a primary key or a foreign key - meaning a Parent-Child data relationship - must be changed everywhere it is used in a consistent way to maintain the referential integrity of the data. Considerations for one-to-one, one-to-many and many-to-one data relationships must be supported.
- **Authorization** - only authorized personnel should be allowed to define the data transformations and all data extracted from live systems must be forced to go through the appropriate transformations unseen in its native form.

The Task of Data Masking and Test Data Creation

Some organizations have attempted to write applications to perform these functions against data in their live systems as a matter of expediency. However, the challenges and considerations associated with creating and masking test data entail expertise and experience that seldom exists entirely within an average development group. Therefore, some consultancy or training is usually required to gain skill in these areas prior to project commencement.

These homegrown applications are nearly always 'requirement specific' as a particular development project only needs their project's data. As a result, each new development project presents entirely different data requirements resulting in these homegrown applications seldom being reusable or extensible without significant modifications. This approach may provide a 'point-specific' solution but leaves many organizations without a corporate-wide strategy for test data and data privacy compliance.

These 'home-grown' utilities incur an opportunity cost since an internal development team must excuse themselves from their primary function of supporting new or existing lines of business. The residual costs continue indefinitely as these applications must be maintained. They also delay any

other project dependent upon the successful implementation of its internal data sub-setting and masking project. Therefore, the initial and hidden cost of internally developed solutions must be weighed against the alternatives. These costs can be significantly higher than a COTS (Commercial Off The Shelf) software solution when all factors are considered.

The development time and financial investment devoted to these limited-scope 'home-grown' applications often exceeds the delivery dates for the projects they purport to help. Whereas a COTS software tool to satisfy these requirements can be justified within a single development project by ensuring that the project is not delayed either by the data privacy requirements or the need to adequately test.

Homegrown data sub-setting and masking utilities are problematic and expensive, particularly if consistency, uniqueness, referential integrity and authorization are to be maintained across an enterprise. These issues coupled with the administrative task of maintaining multiple testing databases causes many organizations to search for a cost effective data masking and test data solution to accelerate their application development, testing and data masking requirements.

SoftBase's TestBase

SoftBase Systems, Inc. offers a comprehensive test data management tool called TestBase. One of TestBase's many features, called [Data Population](#), addresses the issues of data privacy and test data creation head on.

TestBase can be used to:

- Extract referentially intact sets of data from live production environments
- Mask data as it is extracted using 7 available techniques or combinations
- Select data values from any VSAM or QSAM file
- Select data values from any DB2 table or view
- Select data values from values supplied with the product for names, addresses, phone numbers, etc. in sequential or random fashion
- Select data values described by an SQL select statement (DGF View)
- Select random values
- Jumble numeric or character positions
- Transformation tables to lookup real values and supply result values
- Generate test data from scratch for a single table or dataset or a family of referentially related ones, including one-to-one, one-to-many and many-to-one data relationships
- Guarantee only authorized personnel are allowed to define the data masking rules
- Guarantee all extracts of live information have the data masking rules applied
- Provide developers their individual self-service 'SLICE' of data they can refresh at any time

TestBase provides the ability to enforce data masking rules for any user through a unique "MANDATORY MASKING" technique while at the same time ensuring that the requirements of consistency, uniqueness, referential integrity and proper authorization are maintained. This technique coupled with a variety of masking techniques ensures that ANY test data extracted from a

secure production environment satisfies all current data quality and data privacy requirements for test data management.

Data Privacy Team within Test Data Management

The traditional test data management process typically involves having the application testing team request the DBA team to extract production data and then load the data into a testing database. In some cases it is the application development team who loads the data.

The administration of masking rules is a new concept to many organizations. The question that is being asked is who should be responsible for the administration and management of data de-identification and “masking” rules?

With TestBase, organizations have the flexibility to incorporate the administration according to the way the company is organized. If responsibility of test data management belongs to a centralized group, TestBase can provide this type of centralized data management. If the responsibility of test data management is administered by the application DBA team, TestBase can provide this type of centralized data management also.

The overall goal is to minimize the number of people that are authorized to build and maintain masking rules. Because of this, many companies tend to assign data masking responsibilities to the DBA team or test data management team.

Data Privacy Team

Organizations who have data privacy requirements typically establish data privacy teams. The data privacy team is made up of a diverse group of people with different expertise and responsibilities within your corporation. The members on this team may include the following roles:

- **IT Auditor** – IT auditors will want to spot check testing environments to make sure the data privacy policies are being followed. Organizations should have a mechanism to prove data privacy compliance to external auditors. TestBase provides audit reports to ensure that your test data is in compliance with your corporate data privacy policies and directives.
- **Security Officer** – Security officers are responsible for granting access to production data, the use of TestBase, and the authority to populate the test environment.
- **Database Administrator** – The Database Administrator is usually responsible for installing and verifying TestBase. If the organization does not have a Test Data Management Group then the Database Administrators tend to be responsible for maintaining masking rules and the movement of data from production to test.
- **Test Data Management Group (owner of Test Data) TDMG** - The Test Data Management Group is usually formed in larger organizations that have many different business units needing data privacy implemented and are looking to centralize the role. This team will help define and maintain the masking rules along with the responsibility to populate test environments with de-identified production data.

- **Data Privacy Compliance Officer** - The data privacy compliance officer is usually someone who will ensure to upper management that data privacy is being taken seriously and will mentor/coach staff to comply with the corporate privacy policy.
- **DA/BA – Data Analyst or Business Analyst** – The people within the organization who understand the data and how it is used in your business are very important to the success of finding what data needs to be de-identified. The DA or BA will work with the DBA and/or TDMG to identify which columns of data are considered sensitive and should be masked when extracted.
- **Corporate Lawyers** – Corporate lawyers will need to provide direct input and advice on the specific requirements for data privacy.

The assemblage of a data privacy team does not necessarily ensure data privacy compliance within an organization. A plan must be devised, responsibilities delegated, the plan exercised and the results reported.

A Data Privacy Plan

Once the Data Privacy Team is organized, they may not all commence work at once. The workflow of the team is governed by the speed and schedule of the data privacy plan and the scope of its requirements. However, irregardless of the scope, data privacy requirements generally follow these steps:

- **Defining the Legal and Internal Data Privacy Requirements** – What are the general requirements to comply with legal or internal data privacy mandates.
- **Data Analysis** – What specific data must be masked?
- **Business Process Analysis** – What business rules must be considered when masking and how must they be applied?
- **Data Privacy Implementation** – Establishing the specific rules, procedures, and methods of data privacy compliance.
- **Data Privacy Monitoring** – How data privacy compliance is demonstrated.

Data Privacy Requirements

Often a legal opinion must be obtained in order to determine the extent of data privacy compliance. These opinions define the scope of the requirements and may or may not be limited to the use of corporate data for software testing purposes. Corporate lawyers know and understand the respective data privacy laws that their organizations must comply with. As the organizations gain an understanding of what specific data needs to be masked then they will want to confirm the different techniques being used to de-identify the data with their lawyers.

Data Analysis

The process of analyzing corporate data and identifying the data elements such as name, address, phone number that need to be masked can be time consuming. To speed up this process, clients often ask what data should be masked and how should it be masked. Specific requirements may vary which will make generalizations difficult. The variety of data and data types and masking

techniques do not lend themselves to a single masking technique or generalized list of required data elements which will satisfy the needs of all clients. However, there is one general data privacy rule for testing environments – Remove and replace all confidential and sensitive business data from your mainframe testing environment.

Clients may state that they simply want to encrypt the data so they don't have to think about what data masking techniques should be used to mask the data. This is also problematic because of established business rules related to processing based on values of some of these data elements.

Identify Business Processing

Organizations must analyze their data and identify which specific data elements require masking, but also understand if this data has any business logic associated with it. For example, if a social security account number is to be masked and your organization processes an SSN based on the first 3 digits, then there are some business issues to address. This organization may decide that it cannot mask the first 3 digits but can mask the last 6 digits. This is a perfectly acceptable result but must be understood prior to applying a specific masking technique.

In general, the question organizations must ask is, “Does this data need to be masked before I store it in the test environment and if so, how?” For example, if the data contains information that identifies an individual and contains sensitive information like demographics, health records, or financial records then the data in all likelihood needs to be masked before it is stored in the test environment.

Identify Alias Names

As an organization's data is analyzed, there are often data elements with multiple names. For example, data called Employee Number (EMPNO) and data called Responsible Employee Number (RESPEMP) may be found to exist. If RESEMP is the same as an EMPNO, it should be masked the same way. TestBase can automate the task of masking all of these different columns names based on one masking rule.

Source Systems – Mandatory Masking

As data elements are identified, it may be necessary to record the different source systems this element is found on. This will help organizations audit what systems have sensitive data and to ensure that the data is being masked.

TestBase stores the masking rules at a source level and records if the masking rules are mandatory. This means that whenever someone is extracting data from a source, the data will be masked according to the rules defined on that data element (column). With DB2 tables this source is the table creator. Any table containing the column being masked for a given table creator will automatically be masked during the data extraction.

Referential Integrity (RI)

Database referential integrity is the enforcement of business rules in a database. For example, two tables may exist named Employee and Department. The unique id or primary key in the employee

table is employee number (EMPNO). The unique id or primary key in department table is department number (DEPTNO). Perhaps a business requirement exists that says employees must be assigned to an existing department. Therefore if a new employee is being hired for a new department in a company then a new definition of that department must be created before the new employee can be hired and assigned to that department.

Business rules such as these are stored in TestBase as either System defined or Application defined RI rules. System RI is when the rules are stored in the database engine. Application RI is when the rules are stored in business logic in your application. TestBase uses these RI definitions to ensure that RI is maintained during any data movement and masking process. This means that the business rules associated with that data are maintained and supported in the testing environment.

Deciding Which Masking Technique To Use

As an organization's data, business processes and data privacy requirements are analyzed, the next step is to implement the data privacy techniques. TestBase assists at this stage by offering a number of masking techniques. Below is a list some of the common data elements and techniques used to mask data.

Primary Key Columns

TestBase applies data masking rules to the data during the extraction phase of the data population process. TestBase will use the "VALUE" of the data to apply the rules to. Because of this, random number should never be used as a masking technique for a primary key value. If a random value is needed for a key value then the translation option must be used utilizing pre-build the random values.

The two most common techniques used in TestBase to mask a primary key column are Jumble and Translation.

Jumble - The process of moving the bytes of data around.

For example: for a SSN of 559-01-1392, applying a masking rule to transpose the values of one numeric position to another such that the values are rearranged in the following manner:

1 to 9 and 9 to 1

2 to 5 and 5 to 2

3 to 8 and 8 to 3

4 to 7 and 7 to 4

Resulting in: 219-35-1095

Translation - The process of pre-building a look up table with the old (source) values and the new value. TestBase uses this table to generate the new value.

For example: a masking rule may require that the first 3 digits of an SSN always be "000".

Using these 3 examples,

559-01-1392

431-19-3348

312-21-5487

TestBase would utilize the following translation table:

Old/New

559-01-1392 / 000-00-0001

431-19-3348 / 000-00-0002

312-21-5487 / 000-00-0003

When TestBase is processing, the source table is read. The value of 559-01-1392 is used as the key into the translation table. The value returned is 000-00-0001.

Name Columns

When processing names, those names must look real and be reasonably representative for the sample population. If no requirement exists for valid looking names then jumbling the data will suffice. To support masking names with valid names, TestBase provides a [Supplied Values Extended Option](#).

TestBase delivers a supplied value table for last name, male first names, and female first names. These tables contain a sequence number and a value. The supplied value table can be read sequential or random. The preferred method is to use the random option.

For example: Mask the last name using the LNAM supplied value table in random order. Processing: TestBase will read the source table, generate a random number then read the LNAM supplied value table. The random number is used as a look up key value so the names will not come back in sequential order.

Salary Columns

Some organizations require that salary information relating to an employee be masked. TestBase provides a [Random](#) function extended with MIN and MAX values for the random generator.

For example: a masking rule may require that all monthly salary information be randomly masked but can be no less than \$3,000.00 and no more than \$6,000.00

Using these 3 examples,

Salary 1 = \$3,931.92

Salary 2 = \$5,395.18

Salary 3 = \$4,761.55

TestBase would randomly generate the following possible values:

Old/New

\$3,931.92 / \$5,558.24

\$5,395.18 / \$4,786.73

\$4,761.55 / \$5,795.65

Address Columns

The [Supplied Values](#) table extended option is usually used to mask the address data.

For example: a masking rule may require that all address data be masked but the masked data must represent a reasonable address. TestBase provides a [Supplied Values](#) table with addresses that can be substituted for the original address. Additionally, the [Supplied Values](#) table can be populated from external sources for additional needs.

Additional Masking Requirements

While TestBase offers an extensive list of data masking techniques, those listed here are but a few of the most common masking techniques. It may be that specific business requirements and needs may differ from what TestBase provides. TestBase was designed to be a customer extendable product. TestBase supports a column exit, table exit and global exit. The availability of these various exits means that if an organization requires more complex edits to change or mask data, then a COBOL program can be developed which can be called to process this specific data.

If your organization feels that its masking needs are outside the scope of TestBase, please contact SoftBase Systems with your data masking needs for instructions on how to manage your specific masking requirements.

Proving Data Privacy Compliance

Once a Data Privacy Plan has been implemented and all data masking requirements have been satisfied, it must be monitored for compliance. The IT auditor or designated administrator should periodically sample test data and compare it to the original production data to determine if all required data that should have been masked was in fact masked according to the rules established in the Data Privacy Plan. TestBase provides a series of auditor reports which permits this type of examination and verification.

Conclusions

The task of complying with data privacy directives and requirements while also obtaining and maintaining high quality test data for software development purposes can be overwhelming, tedious and problematic. However, if correctly pursued, an investment in data privacy is also an investment in standardized testing procedures, which ultimately improves application quality, deliverability and maintainability. TestBase, by SoftBase Systems is a fully supported enterprise solution that leverages the expertise of DB2 to accelerate the processes of data privacy compliance while providing high quality test data for application development without the time and expense of internal data privacy solutions.