



Test Data Management

Why you, as an organization, should look at Test Data Management

Table of contents

Legislation	3
Reputation Damage.....	3
Cost.....	4
Productivity.....	4
Generating Test Data.....	5
Conclusion	8

There are 4 reasons

why you, as an organization, should look at Test Data Management.



1 Legislation

All over the world, governments have privacy laws. In Canada there is the Personal Information Protection and Electronic Documents Act (PIPEDA). The USA have Sarbanes-Oxley for Data Protection & Integrity, PCI DSS for payment card data security and HIPAA for federal protections regarding personal health information. The European Union has the Data Protection directive which regulates the processing of personal data within the European Union. The Australian Privacy Act of 1988 regulates information privacy. And there are many more.

2 Reputation Damage

Violation of these laws can not only lead to huge fines (millions of dollars), but also to huge reputation damage, which is the second reason for having test data management.

Some examples:

- A breach of the European rules could mean fines of up to two percent of a company's annual turnover, which in the case of Google could mean up to \$800 million.
- Blue Cross Blue Shield of Tennessee (BCBST) agreed to pay the US Department for Health & Human Services (HHS) \$ 1,500,000 on 9 March 2012 when unencrypted computer hard drives containing protected health information (PHI) of over one million individuals were stolen from a leased facility in Tennessee.
- Also, the internet advertising company DoubleClick, that lost 20% of its market value in March 2000 after the storm generated by privacy concerns associated with its acquisition of Abacus Direct, can serve as an example of possible reputation damage.



One shouldn't also forget that generating test data is a recurring process.



3 Cost

A survey indicates that, when creating test data, about two-thirds of organizations make a full copy of their production environment. Not only do full copies consume expensive storage space, but they are also time-consuming, both in computer time and lead time. Besides the disk space, one shouldn't forget the associated management cost of handling these large volumes of test data: examples are back-up and disaster recovery costs.

4 Productivity

If for every test you will use all production data, it is obvious that testing will take longer and will consume more computer time and resources, but also more time of your people. However, most of the time, a smaller, representative sample of the data will be enough for your testing needs. Another advantage of using a representative – smaller – subset of your data, is that it will enhance the quality and outcome of your testing.

Good testing data will allow you to test the right way and most – if not all – business cases. One shouldn't also forget that generating test data is a recurring process. Once this process is automated and standardized, repeating the creation of test data is a simple execution of an existing procedure.

An automated test data generation solution will also free up valuable resources that now are involved in the test data provisioning process. Nowadays, it is mostly an IT-driven process, where IT people not always understand the business needs and consequences. Having an automated, simpler to use solution can involve the right (business) people in the test data generation process, freeing up IT resources.

Furthermore, copying production data does not solve the privacy-related legal issues. You still need to mask or anonymize the data.



Generating Test Data

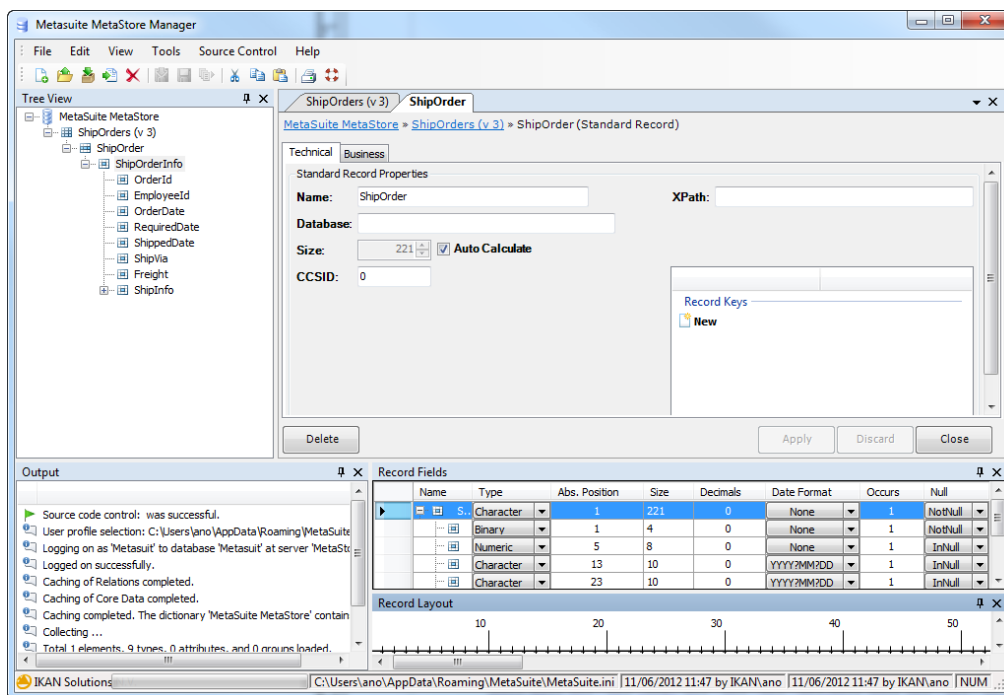
MetaSuite is a GUI-driven data integration tool. After the initial setup, you will just use the MetaStore and the MetaMap component.

MetaStore allows you to collect the data definitions of the source and target structures you want to work on. By collecting the data definitions in the original format, MetaSuite translates these definitions in an internal, uniform format that hides the original (complex) definition to you.

MetaStore provides an hierarchical view of your data files or database tables and gives you full access to the details of your data. Once you have the data definitions, you can use MetaMap to develop your program.

Generating test data is a simple process with MetaSuite:

1. With MetaStore you collect the definitions of your source data in one simple GUI-driven step.
2. With MetaMap you define how you want to sample your data and, by using predefined functions, you can mask the fields you want.
3. Next, you can Generate and run the program with just one-click.

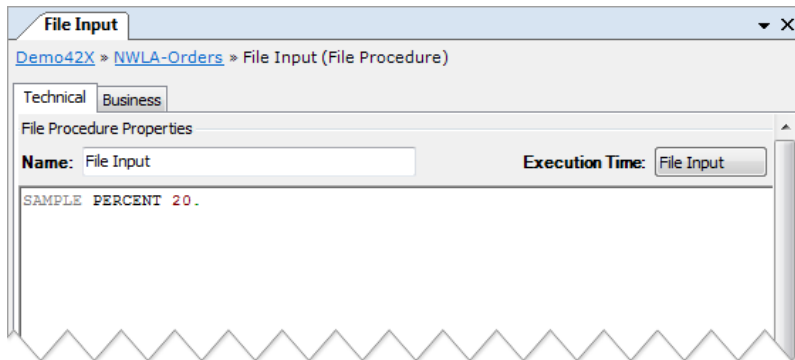


Example of the MetaStore Collect of an XML file:

MetaMap, the second component, allows you to define the subset (sampling) and to mask or anonymize the data. By using the look-a-head-parser, you can select one of the predefined sampling routines.

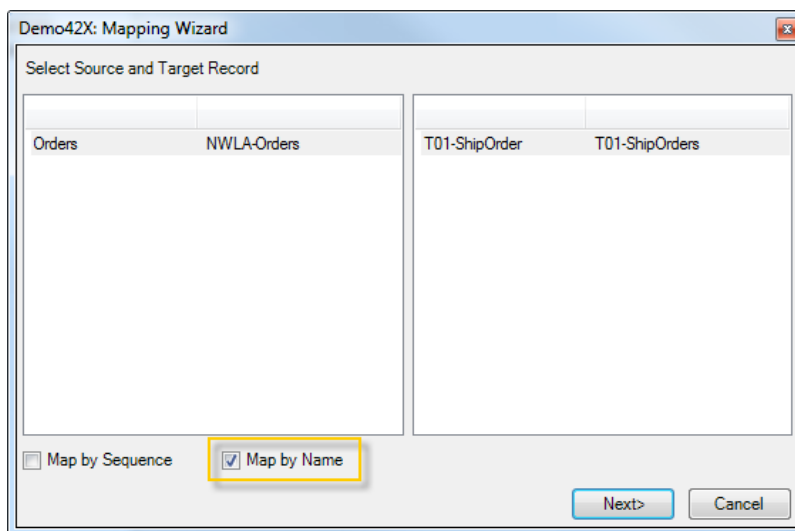
In this example, we just take a sample that represents 20% of the source data, chosen at random. Several other, more complex, sampling routines are available or you could easily create your own sampling command, available for use by everybody by defining a user-defined routine.

1



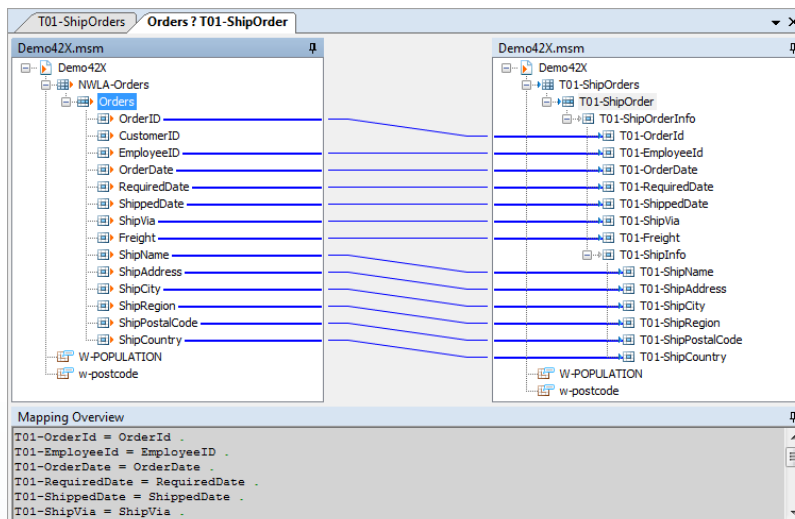
Predefined sample routine that represents 20% of the source data

2



Once you have defined the sample, you just need to map the sample to your target. MetaSuite disposes of a simple Mapping Wizard to do this. The mapping wizard maps by sequence or by name.

3



The result of the sample to target mapping.

4

```

C:\USERS\ANO\MY DOCUMENTS\METASUITE\RUN\DEMO42X.D01
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+
<?xml version='1.0' encoding='ISO-8859-1' ?>
<ShipOrder>
  <ShipOrderInfo OrderId='10250'>
    <EmployeeId>4</EmployeeId>
    <OrderDate>1994-08-08</OrderDate>
    <RequiredDate>1994-09-05</RequiredDate>
    <ShippedDate>1994-08-12</ShippedDate>
    <ShipVia>2</ShipVia>
    <Freight>65.8300</Freight>
    <ShipInfo>
      <ShipName>Hanari Carnes</ShipName>
      <ShipAddress>Rua do Paco 67</ShipAddress>
      <ShipCity>Rio de Janeiro</ShipCity>
      <ShipRegion>RJ</ShipRegion>
      <ShipPostalCode>05454-876</ShipPostalCode>
      <ShipCountry>Brazil</ShipCountry>
    </ShipInfo>
  </ShipOrderInfo>
  <ShipOrderInfo OrderId='10257'>
    <EmployeeId>4</EmployeeId>
    <OrderDate>1994-08-16</OrderDate>
    <RequiredDate>1994-09-13</RequiredDate>
    <ShippedDate>1994-08-22</ShippedDate>
    <ShipVia>3</ShipVia>
    <Freight>81.9100</Freight>
    <ShipInfo>
      <ShipName>HILARIO-Abastos</ShipName>
      <ShipAddress>Carrera 22 con Ave. Carlos Soublette #8-35</ShipAddress>
      <ShipCity>San Cristobal</ShipCity>
      <ShipRegion>Tachira</ShipRegion>
      <ShipPostalCode>5022</ShipPostalCode>
      <ShipCountry>Venezuela</ShipCountry>
    </ShipInfo>
  </ShipOrderInfo>
</ShipOrder>

```

In the next step, you can mask the data. The same simple principle applies: you use one of MetaSuite's pre-defined masking routines or you develop your own user-defined sampling routine. Here's the original input file

5

```

T01-OrderId
Demo42X » T01-ShipOrders » T01-ShipOrder » T01-ShipOrderInfo » T01-OrderId (Target Field)
Technical
Target Field Properties
Name: OrderId
Accumulate
T01-OrderId = SYSTEM-FUNCTION RANDOM-NBR ( 123 , 1234 ) .

```

Edit the target field properties to replace a numeric field (in our example the OrderId field) with a random number between 123 and 1234.

6

```

C:\USERS\ANO\MY DOCUMENTS\METASUITE\RUN\DEMO42X.D01
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+
<?xml version='1.0' encoding='ISO-8859-1' ?>
<ShipOrder>
  <ShipOrderInfo OrderId='627'>
    <EmployeeId>4</EmployeeId>
    <OrderDate>1994-08-08</OrderDate>
    <RequiredDate>1994-09-05</RequiredDate>
    <ShippedDate>1994-08-12</ShippedDate>
    <ShipVia>2</ShipVia>
    <Freight>65.8300</Freight>
    <ShipInfo>
      <ShipName>Hanari Carnes</ShipName>
      <ShipAddress>Rua do Paco 67</ShipAddress>
      <ShipCity>Rio de Janeiro</ShipCity>
      <ShipRegion>RJ</ShipRegion>
      <ShipPostalCode>05454-876</ShipPostalCode>
      <ShipCountry>Brazil</ShipCountry>
    </ShipInfo>
  </ShipOrderInfo>
  <ShipOrderInfo OrderId='515'>
    <EmployeeId>9</EmployeeId>
    <OrderDate>1994-08-12</OrderDate>
    <RequiredDate>1994-09-09</RequiredDate>
    <ShippedDate>1994-08-15</ShippedDate>
    <ShipVia>3</ShipVia>
    <Freight>148.3300</Freight>
    <ShipInfo>
      <ShipName>Richter Supermarkt</ShipName>
      <ShipAddress>Starenweg 5</ShipAddress>
      <ShipCity>Geneve</ShipCity>
      <ShipRegion>' '</ShipRegion>
      <ShipPostalCode>1204</ShipPostalCode>
      <ShipCountry>Switzerland</ShipCountry>
    </ShipInfo>
  </ShipOrderInfo>
</ShipOrder>

```

The resulting file after applying the data masking routines.

Conclusion

It is clear that MetaSuite is easy to use and that you do not require seasoned software programmers to do this. MetaSuite allows people that know the business to generate the test data themselves. Through the intuitive interface and ease of use, they can even try out different scenario's (what ifs) before generating the final program.

More information on how MetaSuite helps with generating test data can be found in the standard user documentation on the product website: www.metasuite.be/infocenter

IKAN Solutions N.V.
Kardinaal Mercierplein 2
2800 Mechelen
Tel. +32 15 797306
info@ikan.be
www.ikan.be

© Copyright 2012 IKAN Solutions N.V.
Metasuite is a trademark of IKAN Solutions N.V.

IKAN