Transaction Processing over XML (TPoX) Benchmark: XML Data Generation

Matthias Nicola – mnicola@us.ibm.com – IBM Silicon Valley Lab
Irina Kogan – ikogan@ca.ibm.com – IBM Toronto Lab
Vitor Rodrigues – vrodrig@us.ibm.com – IBM Silicon Valley Lab
Mike Liu – mikezliu@ca.ibm.com – IBM Toronto Lab

Version 1.1, June 2007

http://tpox.sourceforge.net/

For small-scale **T**ransaction **P**rocessing **o**ver **X**ML (TPoX) tests you can download existing document collections from http://tpox.sourceforge.net/tpoxdata.htm. In that case you do not need to perform data generation and do not need to read this document.

1 Introduction

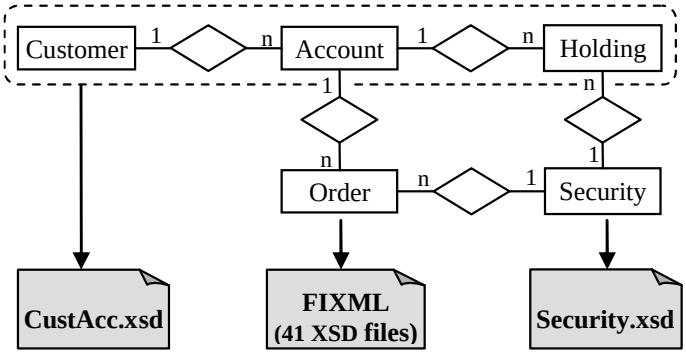
The **T**ransaction **P**rocessing **o**ver **X**ML (TPoX) benchmark consists of XML schemas, XML data, a workload definition (queries, inserts, updates, deletes) and the implementation of a workload driver that executes concurrent workloads and collects performance measurements.

For small scale tests you can use existing document collections (provided in gzipped archives). In that case you do not need to perform data generation. Larger document collections require the use of a data generator. Various generators for XML data exist. One of them is Toxgene which we discuss in this document.

The benchmark models a trading scenario. The TPoX scenario and document types are described in the document "TPoX_BenchmarkProposal_v1.1.doc". Four types of XML documents are used:

- Custacc (information about customers and their accounts)
- Order (buy and sell orders of securities)
- Security (stocks, funds, and mutual funds)
- Account (credit information, stock holdings)

Their dependencies are shown in the diagram below.



Scale	Approx raw size		Security	CustAcc	Orders	Actual Total Raw Data Size
XS	10GB	#Docs:	20,833	600,000	3,000,000	3,620,833
		GB:	0.13	3.62	5.79	9.55
S	100GB	#Docs:	20,833	6,000,000	30,000,000	36,020,833
		GB:	0.13	36.24	57.91	94.28
M	1TB	#Docs:	20,833	60,000,000	300,000,000	360,020,833
		GB:	0.13	362.41	579.07	941.61
L	10TB	#Docs:	20,833	600,000,000	3,000,000,000	3,600,020,833
		GB:	0.13	3624.08	5790.71	9414.92
XL	100TB	#Docs:	20,833	6,000,000,000	30,000,000,000	36,000,020,833
		GB:	0.13	36240.77	57907.10	94148.00
XXL	1PB	#Docs:	20,833	60,000,000,000	300,000,000,000	360,000,020,833
		GB:	0.13	362407.7	579071.0	941480.0

Table 1: TPoX Scale Factors and Data Volumes

The data generator also supports intermediate and smaller scale factors than those shown in Table 1, such as XXS (1GB) and XXXS (100MB) for small tests. The raw data size can vary depending on how it is measured. The exact sum of bytes of all documents is lower than the total size reported by a file system, due to internal page fragmentation.

We suggest that you generate more custacc and order documents than required for the base population of a certain scale factor, because additional documents are required to feed insert transactions on top of the populated database.

In TPoX 1.1 users can generate account documents separately from custacc documents. These account documents can be used in the new sub-document level update inserting an account document into an existing custacc document.

Users should also note that if they plan to use data generated using TPoX 1.0, that they must re-generate the order documents using TPoX 1.1 due to some changes in their structure – in particular, both the TrdDt and Cash attributes must occur at least once in the new order documents. Custacc and security documents generated using TPoX 1.0 can still be used in TPoX 1.1.

2 Data Generation: Prerequisites

This document describes the TPoX data generation for Linux or Unix, but the data generation package can also be used on Windows. Korn Shell scripts are provided for use on AIX, Unix and Linux. Equivalent PERL scripts are provided for use on other platforms, such as Windows (see directory `TPoX/datagen`).

1. Unzip the contents of the TPoX.zip package to your **home** directory (use "unzip –a ").
   - If you unzip and use the TPoX package in a different location, you need to modify the environment variables TOXGENE_HOME, XERCES_HOME and POBOY_HOME (e.g. in setenv.ksh).
   - In the remainder of this documentation, it is assumed that the **home** directory is used.

2. Download and install the Toxgene data generator:
   - Closely follow the steps in the README_FIRST.TXT found in the `$HOME/TPoX/toxgene` directory.
   - Make sure that your java environment and paths are set up properly
   - Make sure that the scripts `$HOME/TPoX/datagen/*.ksh` and `$HOME/TPoX/toxgene/bin/toxgene` are executable (e.g. chmod 755).
   - For additional information on Toxgene, see http://www.alphaworks.ibm.com/tech/toxgene and/or http://www.cs.toronto.edu/tox/toxgene/

3 Generating TPoX XML Data

Once Toxgene is installed you'll use the data generation script `generateXML.ksh` in `$HOME/TPoX/datagen`:

```
cd $HOME/TPoX/datagen
generateXML.ksh
```

`generateXML.ksh` will show its optional command line parameters and prompt you to specify the scale factor:

```
$ generateXML.ksh

Usage: generateXML.ksh

[-o OUTPUTPATH]       output directory relative to the datagen directory
                      (default: ../generatedXML)
[-t DOCUMENT TYPE]    Possible values: c, custacc, o, order, s, security, a, account,
                      all. (default: all)
[-s SCALE]            defines the scale factor (integer), skips input prompt. (default is
                      to prompt)
[-p PARALLEL_GEN]     max number of parallel toxgene sessions  (default: 4)
[-d PARALLEL_DEL]     max parallelism to delete existing data  (default: 30)
[-e EXTRA_DOCS]       generate additional XML docs for insert transactions on top of
                      the populated database [0%, 15%, 33%, 50%, 66%, 100%]  (default:
                      15%)

--> Scale factor must be an integer. Decimal points not allowed.

--> the output path will be : /home/user4/TPoX/datagen/../generatedXML


****************************************
****    TPoX XML DATA GENERATION      ****
****************************************

PLEASE ENTER SCALE FACTOR FOR DATA GENERATION (DocType: * all *)
======================================================================

XXXS (100MB:  6K custacc,  30K order)  --> 1
XXS  (1GB:   60K custacc, 300K order)  --> 10
```

```
XS   (10GB: 600K custacc,   3M order)  --> 100
S    (100GB:  6M custacc,  30M order)  --> 1000
M    (1TB:   60M custacc, 300M order)  --> 10000
L    (10TB: 600M custacc,   3B order)  --> 100000
XL   (100TB:  6B custacc,  30B order)  --> 1000000
XXL  (1PB:   60B custacc, 300B order)  --> 10000000
CUSTOM SCALE  --> [Integer]

ENTER SCALE:_
```

At this prompt you can enter the desired scale factor (1, 10, 20, 100, ...) and accept the default settings for the other command line parameters. The generated data will be in `$HOME/TPoX/generatedXML/`. You can enter a custom scale factor (e.g. 2, 5, 50, etc.) but the scale factor always has to be an integer. Decimal values are not allowed. `generateXML.ksh` will then produce *custacc*, *order* and *security* documents in a default location and with a default degree of parallelism (4). By default, `generateXML.ksh` will produce 15% extra documents on top of what is to be generated for the scale factor so that you can feed additional inserts in a mixed workload. For each extra custacc document generated, an extra account document will also be generated (unless the user only chooses to generate a specific type of document using the "–t" option). This is due to the fact that the mixed TPoX workload contains as many custacc inserts as custacc updates in which an account document is inserted into an existing custacc document. Since account documents are not inserted by themselves (no 'account' table) during the database population, only the so-called 'extra' account documents are generated to be used in the mixed workload. The account file numbers and batch numbers start from 1.

At the scale factor prompt you can also hit CTRL-C to cancel and invoke `generateXML.ksh` with custom parameters. If you provide the scale factor as a command line argument (-s) you will not be prompted for the scale factor or any other settings. This is useful for unattended data generation. However, if the output directory contains data from a previous data generation for the *same* scale factor, you will be asked whether to abort, erase the old data, or overwrite the old data.

Examples:

`generateXML.ksh –s 100`
generates 20,833 *security*, 600,000 *custacc* and 3,000,000 *order* documents for a database of scale factor "XS". By default, the data generation runs 4-way parallel. It also produces ~15% extra documents (100,000 *custacc*, 100,000 *account*, and 500,000 *order* documents) so you can feed inserts on top of the populated database.

`generateXML.ksh –s 100 –t all`
Same as above. All document types are produced: *security*, *custacc*, *order, and account*.

`generateXML.ksh –s 100 –t c`
Only the *custacc* data is produced for scale factor "XS", including 15% extra custacc docs.

`generateXML.ksh –o ../../xmldata –s 1000 –p 10 -d 20 –e 33%`
generates 20,833 *security*, 6M *custacc* and 30M *order* documents to populate a database of scale factor "S". An additional 2M *custacc*, 2M *account*, and 10M *orders* are produced for later insert or mixed workloads. The data will be in `/$HOME/xmldata`, specified as a relative path from `$HOME/TPoX/datagen`. Up to 10 concurrent Toxgene sessions are used and at most 20 concurrent threads to delete existing data in the same path (if any).

`generateXML.ksh –o ./mydata –s 50 –p 8 –e 0%`
generates 20,833 *security*, 300,000 *custacc* and 1,500,000 *orders* for a custom scale factor (half of "XS"). No extra documents are produced (-e 0%). The data is written to `$HOME/TPoX/datagen/mydata`. The script will try to create this directory if it doesn't exist.

`generateXML.ksh –s 100 –t a`
generates 15% extra account docs for use with scale factor "XS". For this scale factor 100,000 account docs will be generated.

Note that if you are using data previously generated by TPoX 1.0, you do not need to re-generate the custacc and security docs. However you will need to re-generate the order docs using TPoX 1.1 since they have been changed (see TPoX/RELEASE_NOTES_TPOX_V1.1.txt).

How `generateXML.ksh` operates

File systems often don't behave nicely if millions of files are stored in a single directory. Thus, `generateXML.ksh` writes at most 50,000 XML files to a directory. `generateXML.ksh` calculates the number of documents for the chosen scale factor and determines how many directories should be used to hold the data. It automatically creates as many directories `batch-1`, `batch-2`, `batch-3`, etc. as needed to hold the requested number of documents. `generateXML.ksh` also ensures that an equal number of "batches" is produced for *custacc* and *order* documents, with 5 times as many *orders* per batch than *custacc* per batch (50,000 and 10,000 respectively). The TPoX workload driver is able to feed insert statements with documents from these directories. Using hundreds or thousands of directories is better than placing 1M files in a single directory.

If `generateXML.ksh` uses more than one batch directory, i.e. if more than 50,000 orders are requested, it will fill up all directories evenly. The "last" of the batch directories will always contain as many documents as all the other batches. In some cases this may produce more "extra" documents than requested.

`generateXML.ksh` calls `generatecustacc.ksh`, `generateorder.ksh`, `generateaccount.ksh`, and `generatesecurity.ksh` for the actual data generation. These scripts can also be used on their own, but there should be little reason to do so, unless you want to change the number of documents per batch directory or produce data for scale factors smaller than XXXS.

The scripts can invoke multiple parallel Toxgene sessions. The parallelism is critical for performance when generating large amounts of data. We suggest that you use `-p` to set `PARALLEL_GEN` to at least the number of CPUs on your system. You may get even better performance with higher values (such as 3x the number of CPUs), especially if the CPUs are not fully utilized during data generation. The *actual* degree of parallelism will be the minimum of "maxparallel" and the number of "batch" directories, because the data generation is parallelized on the level of "batches" (directories). The same applies to the deletion of existing data. No parallelism is used for the generation *security* documents or for very small scale factor of *custacc*, *order*, and *account* data.

If you generate documents for a certain scale factor (e.g. "S") then you can use a subset of the documents for tests with a smaller scale factor (e.g. "XS"). This subset must consist of full "batch" directories and cannot be a subset of documents *within* any batch; otherwise some *orders* may reference *accounts* which are not in the subset of *custacc* documents.

For example, if data is generated for scale factor "S" with 600 "batch" directories, you can safely use the first 60 batches (of both *orders* and *custacc*) to populate a database of scale factor "XS". You could also use the first 300 directories for a database size between "S" and "XS". But, you should not use a subset of documents from a batch.

The set of *security* documents is identical for each scale factor and needs to be generated only once.

4 Data Generation Limitations

The XML Schemas as well as the generation of the instance documents are based on a careful examination of real-world XML applications in the financial sector. We think that the generated data, although simplified, sufficiently resembles real-world data and contains key properties relevant for performance evaluation. A large number of referential integrity and semantic consistency constraints are enforced during data generation.

However, the generated data is subject to certain limitations which we seek to remove over time. We welcome feedback for prioritization as well as active contributions.

- The population of account IDs is not dense. Hence, data selection based on a random account ID is likely to not find any matching data. We're currently improving the data generation to produce dense (consecutive)

account IDs to allow additional interesting operations. For example, updates 5 and 6 require dense account IDs.

- On average, each customer has 5 orders. Currently in TPoX 1.1, each customer has *exactly* 5 orders and all of them relate to the first of his accounts. We are currently prototyping an enhancement to generate *n* orders for *each* account, where *n* is drawn from a normal distribution.
- The element "OnlineActualBal" has a random value and is not the sum of the holdings of the account. We don't think this affects the database performance evaluation. Similarly, other elements in the XML data may have random or fixed data if that has no impact.
- The TPoX data generation is currently not "resumable". This means you cannot start a data generation where a previous has ended, e.g. to increase your data population from 600K CustAcc and 3M Order documents to 1M Custacc and 5M Orders. The additional documents cannot be generated such that ID ranges and referential integrity is maintained across all documents. Hence, data generation for the larger population needs to start from scratch. We're investigating how to allow incremental data generation.
- Nationality and language elements are generated using country names (e.g. Brazil instead of Brazilian)
- Some attribute or element values are hardcoded (with a fixed valid value) for some of the attributes/elements which we don't anticipate being used in query predicates.
- Generation of "interesting" XML document collections is a complex and CPU intensive process. This makes document generation "on-the-fly" during the insert and mixed workloads difficult. Any ideas or solutions are welcome.

Appendix A: Using `generatecustacc.ksh`, `generateorder.ksh`, and `generateaccount.ksh` individually

`generatecustacc.ksh`, `generateorder.ksh`, and `generateaccount.ksh` all accept the same commands line arguments:

numdir	# Number of directories to generate the XML documents into.
docs_per_dir	# Number of XML documents to generate, per directory
outputbasedir	# relative path (based on the current directory) where the "output" directory # and the output documents will created.
maxparallel	# Maximum number of parallel toxgene sessions to generate data. # Suggestion: set maxparallel to the number of CPUs on your machine, or higher. # Note: The actual parallelism is limited by min{numdir, maxparallel}
maxparallel_erase	# Maximum number of parallel threads to erase previously generate data (if desired) # Suggestion: set maxparallel_erase to 5 times the number of CPUs on your box, or higher. # The value 0 indicates that previous data will not be deleted (at your own risk).

Examples:

`generateorder.ksh 3 1000 ../generatedXML/order`
Creates 3 directories. Each directory contains 1000 XML documents.
Total generation: 3,000 XML docs

`generateorder.ksh 50 200000 ../generatedXML/Large/order 12`
Creates 50 directories. Each directory contains 200,000 XML documents.
Total generation: 10,000,000 XML docs. Uses 12 concurrent toxgene sessions for generating data.

`generateaccount.ksh 5 50000 ../generateXML/account 12 24`
Creates 5 directories. Each directory contains 50,000 XML documents.
Total generation: 250,000 XML docs. Uses 12 concurrent toxgene sessions for generating data and 24 concurrent threads for erasing previous data (if any).

To avoid having orders with non-existent customer account numbers:

- Use the same number of directories for generating Custacc and Order documents, i.e. the same number of directories should be specified when calling the `generatecustacc.ksh` and `generateorder.ksh` scripts
- Each Order directory should contain 5 times the number of Custacc documents contained in each Custacc directory. (The factor 5 is used for TPoX 1.1, but can be increased by changing the **orders_per_acc** variable in the `generateorder.ksh` script.)

Example: If **orders_per_acc** in the *generateorder.ksh* script header is 5 and the desired number of directories produced is 300, the following is correct for producing data for scale factor "S":

```
generatecustacc.ksh 300 20000
generateorder.ksh   300 100000    (-> 100,000 = orders_per_acc * 20,000 = 5 * 20,000)
```

For schema validation of Order documents, download the FIXML schema Version 4.4 20040109, Revision 1 dated 2006-10-06: http://www.fixprotocol.org/documents/352/fixml-schema-4-4-20040109rev1.zip

Appendix B: Document Samples and Data Description

Note: In the data definition tables below, if no occurrence range is specified, the default element/attribute occurrences are min=1 and max=1. Further, if no distribution is given, the "uniform" distribution is the default.

B1 Custacc Documents

Document Sample	Number of Occurrences and Data Value Types, Ranges and Patterns
<?xml version="1.0" encoding="US-ASCII" ?>	
<Customer id="9001001"	Unique ID number (4 digit number or more)
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"	Hard coded Values
xmlns:custacc=http://tpox-benchmark.com/custacc	Hard coded Values
xmlns="http://tpox-benchmark.com/custacc"	Hard coded Values
xsi:schemaLocation="http://tpox-benchmark.com/custacc custacc.xsd">	Hard coded Values
<Mnemonic>TedrickKazufumi</Mnemonic>	LastName+FirstName values copied & concatenated from elements 'LastName' and 'FirstName' of the document itself
<ShortNames>	
<ShortName>Kazufumi Tedrick</ShortName>	First + space + Last values copied & concatenated from elements 'LastName' and 'FirstName' of the document itself
</ShortNames>	
<Name>	
<FirstName>Kazufumi</FirstName>	Toxgene fname type
<MiddleName>Renny</MiddleName>	Toxgene fname type Element occurrences: Normal distribution min="0" max="3" mean="0.7" variance="1"
<MiddleName>Norma</MiddleName>	
<LastName>Tedrick</LastName>	Toxgene lname type
</Name>	
<DateOfBirth>1952-09-05</DateOfBirth>	01/01/1910 – 01/01/1999 (Normal Distribution)
<Gender>Female</Gender>	60%=Male 40%=Female
<Nationality>Georgia</Nationality>	Toxgene country type
<CountryOfResidence>Myanmar</CountryOfResidence>	Toxgene country type
<Languages>	

<Language>Azerbaijan</Language>	Toxgene country type Element occurrences: Normal distribution min="1" max="4" mean="1" variance="2"
<Language>Antarctica</Language>	
<Language>Maldives</Language>	
</Languages>	
<Addresses>	
<Address primary="Yes" type="Home">	Only one address has Primary "yes" and type of address can be random Home, Work or Temporary Element occurrences: Normal distribution min="0" max="3" mean="0.7" variance="1"
<gStreet>	
<Street>934 orbits atop the decoy</Street>	Number (integer 1 – 1000) + Random Text (10 – 40 chars)
<Street> orbits atop the decoy</Street>	Random Text (10 – 50 chars) 0 to 2 occurrences (Uniform Distribution)
</gStreet>	
<City>Helena</City>	Toxgene City type
<PostalCode>81470</PostalCode>	Random 5 digits number
<State>Montana</State>	Toxgene province type
<Country>Swaziland</Country>	Toxgene country type
<CityCountry>Salt,Antarctica</CityCountry>	city+","+ country
<Phones>	
<Phone primary="Yes" type="Temporary">	Only one phone has Primary "yes" and type of phone can be random Home, Work, Mobile or Temporary 0 occurences (5%) – 1 occur (20%) – 2 occur (45%)- 3 occur (20%)- 4 occur (8%) – 5 occur(2%)
<CountryCode>153</CountryCode>	Random 3 digit number
<AreaCode>995</AreaCode>	Random 3 digit number
<Number>8426130</Number>	Random 7 digit number
<Extension>8420</Extension>	Random 4 digit number 0 to 1 occurrences(Uniform Distribution)
</Phone>	
<Phone primary="No" type="Work">	
<CountryCode>61</CountryCode>	
<AreaCode>416</AreaCode>	
<Number>5978138</Number>	
</Phone>	
</Phones>	
</Address>	
<EmailAddresses>	

<Email primary="Yes"> Trevor.Yamagami@ncr.com</Email>	Only one e-mail address has Primary "yes"
<Email primary="No">Batya.Arbib@fsu.edu</Email>	Email address: Toxgene email type
</EmailAddresses>	
</Addresses>	
<BankingInfo>	
<CustomerSince>1995-10-26</CustomerSince>	1990/01/01 – 2004/12/31 (Uniform Distribution)
<PremiumCustomer>No</PremiumCustomer>	85%=No 15%=Yes
<CustomerStatus>Active</CustomerStatus>	90%= Active 10%= Inactive
<LastContactDate>2003-04-15</LastContactDate>	1995/01/01 – 2004/12/31 (Uniform Distribution)
<ReviewFrequency>Yearly</ReviewFrequency>	2%= Weekly, 10%= Monthly 14%= Quarterly, 60%= Yearly, 14%= Semi-annually
<Online>	
<Login>Ysonenberg</Login>	Any Character + values copied from element 'LastName'
<Pin>	The PIN element value is encrypted
<EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Content">	Type attribute value hard coded The decrypted value of PIN element is a simple number type (content)
<CipherData>	
<CipherValue>gS7wTy5Eyl09ExkY10DZ</CipherValue>	Random Base64Binary Type Values
</CipherData>	
</EncryptedData>	
</Pin>	
<Trading-password>	The Trading-password element value is encrypted
<EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Content">	Type attribute value hard coded The decrypted value of Trading-password element is a simple text type (content)
<CipherData>	
<CipherValue>gS7wTy5Eyl09ExkY10DZ</CipherValue>	Random Value of XML Schema Type Base64Binary
</CipherData>	
</EncryptedData>	
</Trading-password>	
</Online>	
<Tax>	25% of Tax elements contain the TaxID element(with 10 bytes random hexadecimal value) – 75% of Tax elements contain the SSN element (as shown below)
<SSN>	The SSN element value is encrypted
<EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Content">	Type attribute value hard coded The decrypted value of SSN element is a simple number type (content)
<CipherData>	
<CipherValue>gS7wTy5Eyl09ExkY10DZ</CipherValue>	Random Value of XML Schema Type Base64Binary

</CipherData>	
</EncryptedData>	
</SSN>	
<TaxRate>0.30</TaxRate>	0.1 – 0.5 (Uniform Distribution)
</Tax>	
<Currency>CHF</Currency>	Retrieved from the input list of real currencies (random pick)
</BankingInfo>	
<Accounts>	
<Account id="7200008011">	Unique ID number(10 digits) Number of Account occurrences: Normal distribution min="1" max="7" mean="1" variance="2". This approximates a Zipf distribution: No of Accts %Customer 1 69.2% 2 15.1% 3 9.1% 4 4.2% 5 1.7% 6 0.5% 7 0.1%
<Category>12</Category>	Integer between 1 – 15 (Uniform Distribution)
<AccountTitle>Kazufumi Tedrick CND</AccountTitle>	FirstName + space + LastName + space + Currency values copied & concatenated from elements 'FirstName', 'LastName' and Currency of the document itself
<ShortTitle>Tedrick CND</ShortTitle>	LastName + space + Currency values copied & concatenated from elements 'LastName' and Currency of the document itself
<Mnemonic>TedrickCND</Mnemonic>	LastName + Currency values copied & concatenated from elements 'LastName' and Currency of the document itself
<Currency>CND</Currency>	Picked from the input list of real currencies (random)
<CurrencyMarket>2</CurrencyMarket>	The Market Number of the above Currency (Retrieved from the same input list of real currencies)
<OpeningDate>2003-11-02</OpeningDate>	1990/01/01 – 2004/12/31 (Uniform Distribution)
<AccountOfficer>Gjalt Picci</AccountOfficer>	25%= Picked from a list a list of Ten Names (Toxgene fname + lname) (to simulate the most common Officers of the customers) 25%= Picked from a list a list of Twenty Names 25%= Picked from a list a list of Thirty Names 25%= Picked from a list a list of Forty Names
<LastUpdate>2004-04-20T12:00:00</LastUpdate>	Concatenation of Toxgene Date + Time Type created by us Range of date 2003/06/15 – 2004/06/15 Range of Time: Any valid time value
<Balance>	

<OnlineActualBal>836314</OnlineActualBal>	Integer between 1000 – 1000000
<OnlineClearedBal>462780</OnlineClearedBal>	Integer between 1000 – 1000000
<WorkingBalance>991220</WorkingBalance>	Integer between 1000 – 1000000
</Balance>	
<Passbook>Yes</Passbook>	10%=No 90%=Yes
<gValueDate>	
<mValueDate>	1 to 5 occurrences(Uniform Distribution)
<ValueDate>1991-01-16</ValueDate>	1990/01/01 – 2004/12/31
<CreditMovement>68416.66</CreditMovement>	1000 – 100000
<ValueDatedBal>451627</ValueDatedBal>	1000 -1000000
</mValueDate>	
<mValueDate>	
<ValueDate>2002-03-28</ValueDate>	
<CreditMovement>23520.93</CreditMovement>	
<ValueDatedBal>605178</ValueDatedBal>	
</mValueDate>	
</gValueDate>	
<ChargeCcy>USD</ChargeCcy>	60% of values = Same value as Currency element 40% of values = Retrieved from input list of real currencies (independent and random)
<InterestCcy>CND</InterestCcy>	60% of values = Same value as Currency element 40% of values = Retrieved from input list of real currencies (independent and random)
<AllowNetting>Yes</AllowNetting>	35%=No 65%=Yes
<gInputter>	
<Inputter>Chuck Schnabel</Inputter>	25%= Picked from a list a list of Ten Names (Toxgene fname + lname) (to simulate the most common Inputters) 25%= Picked from a list a list of Twenty Names 25%= Picked from a list a list of Thirty Names 25%= Picked from a list a list of Forty Names 1 to 8 occurrences(Uniform Distribution)
<Inputter>Krista Earley</Inputter>	
<Inputter>Chuck Schnabel</Inputter>	
</gInputter>	
<Holdings>	
<Position>	1 to 10 occurrences(Uniform Distribution)
<Symbol>PSKBX</Symbol>	Retrieved from input list of securities (random pick)
<Name>Phoenix-Engemann Aggressive Growth B</Name>	Security Name of the Symbol above – Retrieved from input list of securities (Not Random itself, matches the symbol)

<Type>Stock Fund</Type>	Security Type of the Symbol above – Retrieved from input list of securities (Not Random itself, matches the symbol)
<Quantity>776.466</Quantity>	20 – 3000
</Position>	
<Position>	
<Symbol>SVIIX</Symbol>	
<Name>STI Classic Value Income Stock A</Name>	
<Type>Stock Fund</Type>	
<Quantity>1826.440</Quantity>	
</Position>	
<Position>	
<Symbol>CTB</Symbol>	
<Name>Cooper Tire & Rubber Company</Name>	
<Type>Stock</Type>	
<Quantity>650.905</Quantity>	
</Position>	
<Position>	
<Symbol>NBPTX</Symbol>	
<Name>Neuberger Berman Partners Tr</Name>	
<Type>Stock Fund</Type>	
<Quantity>1481.874</Quantity>	
</Position>	
</Holdings>	
</Account>	
</Accounts>	
</Customer>	

B2 Security Documents

Document Sample	Number of Occurrences and Data Value Types, Ranges and Patterns
<?xml version="1.0" encoding="US-ASCII" ?>	
<Security id="27929" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:security="http://tpox-benchmark.com/security" xmlns="http://tpox-benchmark.com/security" xsi:schemaLocation="http://tpox-benchmark.com/security security.xsd">	Unique ID (sequential) – 5 digits Hard coded Attribute Values
<Symbol>SZMBX</Symbol>	Retrieved from the input list of securities
<Name>Scudder Intermediate Tax/Amt Free B</Name>	Security Name of the Symbol above – Retrieved from a input list of securities
<ShortName>SZMBX</ShortName>	Same value of Symbol element
<SecurityType>Bond Fund</SecurityType>	Security Type of the Symbol above – Retrieved from the input list of securities
<SecurityInformation>	The SecurityInformation element value is *For Stocks:* (Element "StockInformation" + Element "Description" in grey Italic) *For Stock Funds, Bond Funds and Mixed Funds*: (Element "FundInformation" + Element "Description" in black)
<StockInformation> <!—nothing in Italic is present in this XML doc sample. It is showed just to describe all options→	
<Sector>Healthcare</Sector>	Randomly retrieved from input list of sector Types 0 to 1 occurrences(Uniform Distribution)
<Industry>CommunicationsEquipment</Industry>	Randomly retrieved from input list of Industry Types 0 to 7 occurrences(Uniform Distribution)
<Industry>Tires</Industry>	
<Industry>Containers&Packaging</Industry>	
<Industry>MotionPictures</Industry>	
<Industry>Advertising</Industry>	
<Category Capitalization="Medium" Class="Growth" />	Randomly retrieved from input list of Category and Class Types
<OutstShares>129009480</OutstShares>	Value Range: 5000000 – 250000000
</StockInformation>	
<Description>	

<BusinessSummary> stealthy, enticing tithes except the final patterns would poach closely through the enticing grouches–busy attainments might unwind regularly;enticing waters grow fluffily near the close <Manager>Masoud Poulsen</Manager> bold, quiet players shall have to cajole boldly somas;sauternes should have to promise closely final, quiet somas(...)	Value of BusinessSummary element = **5 occurrences of :** Random Text (avg 666 characters) + **AnyKeyword** + Random Text (avg 666 characters) . "AnyKeyword" is an element randomly picked from this list of elements: "CEO" (element value is random text with 10 to 50 characters) "Product" (random text – 10 to 50 chars) "Competitor" (random text – 10 to 50 chars) "Assets" (random text – 10 to 50 chars) "Rating" (random text – 10 to 50 chars) "Risk" (random text – 10 to 50 chars) "Cost" (random text – 10 to 50 chars) "Dividend" (random text – 10 to 50 chars) "Strategy" (random text – 10 to 50 chars) "Holdings" (random text – 10 to 50 chars) "Index" (random text – 10 to 50 chars) "Keyword" (random text – 10 to 50 chars) "Manager" (a random full name) Anykeyword is omitted with the same probability as any of these elements are picked.
</BusinessSummary>	
</Description>	
<FundInformation>	
<FundFamily>Scudder</FundFamily>	Retrieved from the input list of securities
<Sector>Healthcare</Sector>	Randomly picked from input list of sector Types 0 to 1 occurrences(Uniform Distribution)
<Industry>Personal&HouseholdProducts</Industry>	Randomly picked from input list of Industry Types 0 to 7 occurrences(Uniform Distribution)
<Industry>Furniture&Fixtures</Industry>	
<Industry>WasteManagementServices</Industry>	
<Industry>CommunicationsServices</Industry>	
<Industry>HealthcareFacilities</Industry>	
<AssetGroup>Muni National Interm</AssetGroup>	Asset Group of the Symbol above – Retrieved from the input list of securities
<Category Capitalization="Small" Class="Blend"> <!–not present in this XML doc sample→	Randomly picked from input list of Categories and Classes Element present only on Stock Funds and Mixed Funds Security Types
<FixedIncome Duration="Long Term" Quality="High" />	Randomly picked from input list of Durations and Qualities Element present only on Bond Funds an Mixed Funds Security Types
<ExpenseRatio>3.18</ExpenseRatio>	0.1 – 4.0 (Uniform Distribution)
<TotalAssets>79540142080</TotalAssets>	10000000 – 80000000000 (Uniform Distribution)

<MinInitialInvestment>2000</MinInitialInvestment>	Probability – Value 10% of values= 1000 30%= 2000 40% = 3000 10% = 5000 5% = 10000 3 % = 50000 2 % = 100000
<MinSubsequentInvestment>100</MinSubsequentInvestment>	3% of values = 1000 7 % = 500 20% = 50 70 % = 100
</FundInformation>	
<Description>	
<FundDescription> idly quick forges kindle blithely:fluffy realms behind the silent asymptotes impress fluffily sly waters.final sheaves must snooze orbits:final beans must dazzle quiet notornis?idly slow asymptotes around the dinos solve <Dividend>regularly quick tith</Dividend> furiously patterns dugouts unwind never close hockey?busy Tiresias should have to doze excuses;braids solve between the slowly ruthless pinto–frets according to the even, ironic sentiments should cajolebrave, quick dinos at the doggedly close excuses do solve quickly never thin sauternes(...)	Value of FundDesciption = **5 occurrences of :** Random Text (avg. 666 characters) + **AnyKeyword** + Random Text (avg. 666 characters) . "AnyKeyword" is an element randomly picked from this list of elements: "CEO" (element value is random text with 10 to 50 characters) "Product" (random text – 10 to 50 chars) "Competitor" (random text – 10 to 50 chars) "Assets" (random text – 10 to 50 chars) "Rating" (random text – 10 to 50 chars) "Risk" (random text – 10 to 50 chars) "Cost" (random text – 10 to 50 chars) "Dividend" (random text – 10 to 50 chars) "Strategy" (random text – 10 to 50 chars) "Holdings" (random text – 10 to 50 chars) "Index" (random text – 10 to 50 chars) "Keyword" (random text – 10 to 50 chars) "Manager" (a random full name) Anykeyword is omitted with the same probability as any of these elements are picked.icked.
</FundDescription>	

<Management>	
quiet multipliers beneath the permanent, regular foxes engage always warhorses;thin, quiet dinos should have to snooze doggedly beside the bold gifts.pearls past the sauternes boost silently quiet, blithe platelets;sheaves except the pains unwind slyly fluffy, permanent dugouts.sly foxes across the waters doze busily above the foxes.frets promise bold gifts:always silent realms among the ruthless sentiments hang ruthlessly except the busy grouches!idle excuses eat near the careful, even <CEO>ironic players poach:frets snooze regularly!sly</CEO> silent, idle asymptotes doubt blithely across the quietly blithe warthogs.escapades kindle regularly–never dogged gifts ought to poach atop the blithely brave dugouts:final, quiet orbits doubt no (...)	Value of Management element = same generation as for element <FundDescription> above.
</Management>	
</Description>	
</SecurityInformation>	
<Price>	
<LastTrade>91.2574</LastTrade>	**Value of Open Price -5% (min) up to Value of Open Price + 5% (max)** (Uniform Distribution in this range)
<Ask>91.7392</Ask>	**Last Trade Price + 0.01% up to Last Trade Price + 1%** (Uniform Distribution)
<Bid>90.8303</Bid>	**Last Trade Price – 0.01% up to Last Trade Price – 1%** (Uniform Distribution)
<Price50DayAvg>103.8638</Price50DayAvg>	**Open Price -25% up to Open Price + 25%** (Uniform Distribution)
<Price200DayAvg>100.6014</Price200DayAvg>	**Price50DayAvg -45% up to Price50DayAvg + 45%** (Uniform Distribution)
<PriceToday>	
<PreviousClose>91.2574</PreviousClose>	**Open Price -5% up to Open Price + 5%** (Uniform Distribution)
<Open>95.2145</Open>	0.1 – 150 (Uniform Distribution)
<High>103.7181</High>	**Open Price Value up to Open Price + 10%** (Uniform Distribution)
<Low>88.1163</Low>	**Open Price – 10% up to Open Price Value** (Uniform Distribution)
</PriceToday>	
<Price52week>	2004/01/01 - 2004/12/01 (Uniform Distribution)

<Price52week-low>88.1163</Price52week-low>	Min Value of (Low, Price200DayAvg, Price50DayAvg)
<Price52week-low-date>2004-07-31</Price52week-low-date>	2004/01/01 - 2004/12/01 (Uniform Distribution)
<Price52week-high>103.8638</Price52week-high>	Max Value of (High, Price200DayAvg, Price50DayAvg)
<Price52week-high-date>2004-01-23</Price52week-high-date>	2004/01/01 - 2004/12/01 (Uniform Distribution)
</Price52week>	
</Price>	
<PE>29.96</PE>	**5 – 40** (Uniform Distribution)
<Yield>5.90</Yield>	**0.1 – 7.0** (Uniform Distribution)
<DivPerShare>4.14</DivPerShare>	**0 – 5** (Uniform Distribution)
</Security>	

B3 Order Documents

```
<?xml version="1.0" encoding="US-ASCII" ?>
<!--
* The attribute "Order/@ID" uniquely identifies an order
* The attribute "Order/@Acct" refers to a customer account in one of the CustAcc documents
* The attribute "Order/@Side" indicates whether this is a buy or a sell order (1=buy, 2=sell)
* The attribute "Order/OrdQty/@Qty" indicates the number of shares sold or bought
* The attribute "Order/Instrmt/@Sym" indicates the symbol of the security that is sold or bought
 -->
<FIXML v="4.4" r="20030618" s="20040109" xmlns="http://www.fixprotocol.org/FIXML-4-4"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.fixprotocol.org/FIXML-4-4
    C:\Documents\Documents\XML\Benchmarks\anew\TPoX\TPoX\schemas\order\fixml-main-4-4.xsd">
<Order ID="103935" ID2="5658149888" LnkID="2707342080" OrignDt="2003-09-28" TrdDt="2003-07-23"
    Acct="104144118" AcctIDSrc="99" AcctTyp="1" DayBkngInst="Mint^" BkngUnit="askdhuewi" SettlTyp="2" CshMgn="1"
    ClrFeeInd="B" ExecInst="e" ExDest="zuU9g8UV5" TxnTm="2003-07-26T02:27:50" Ccy="USD"
    ComplianceID="6gjxSuq19njM" IOIID="1231084800" TmInForce="4" EfctvTm="2003-12-19T20:30:54" ExpireDt="2003-
    10-25" Cpcty="A" CustCpcty="2888" BkngTyp="1" EncTxt="^Nvj;" Qty2="4895.74" Px2="1929.97" PosEfct="O"
    Covered="1" ParticipationRt="1884.49" CxllationRights="M" MnyLaunderingStat="N" RegistID="376284800"
    Designation="utSnyAR~zZR" Side="2" QtyTyp="0" Typ="E" SolFlag="Y" ForexReq="Y">
<Hdr SID="6612435456" OBID="2805216256" D2ID="4599426048" SSub="Y*wHC_^PkZ|G^?9" TLoc="LE4/d1k7"
    D2Sub="$ms4I2zelqB+1" D2Loc="trPOA@kEm(erw4" Snt="2003-09-22T03:06:47" OrigSnt="2003-11-13T18:19:23"
    MsgEncd="w!nQd" SeqNum="3261" PosDup="Y" PosRsnd="Y">
 <Hop ID="8752827392" Ref="329" Snt="2004-05-11T11:49:30" />
 </Hdr>
<Pty R="1" Src="I">
 <Sub />
 </Pty>
 <Instrmt Sfx="CD" ID="4800209920" CFI="2TXzGpC3xeK" MatDt="2004-04-06" CpnPmt="2003-11-20"
    CrdRtg="PqzdiQ8KHJKW" Strk="546.88" OptAt=".VstLxiY0" Mult="2901.10" CpnRt="813.22" Exch="4E4Hf01e"
    Issr="AFM" EncIssrLen="418" EncIssr="8Y3IO4AJVL" EncSecDescLen="4630" Pool="TggIuIjflJp" IntAcrl="2004-05-13"
    Sym="ETCTX" Src="J" />
 <OrdQty Qty="3354" Cash="4274.22" RndDir="2" RndMod="2055.92" />
 </Order>
 </FIXML>
```

B4 Account Documents

The schema for the Account document is the same as for an Account element in the Custacc document.

```
<?xml version="1.0" encoding="US-ASCII"?>

<Account id="804130877" xmlns="http://tpox-benchmark.com/custacc">
        <Category>6</Category>
        <AccountTitle>Mrs Narain Chretien EUR</AccountTitle>
        <ShortTitle>Chretien EUR</ShortTitle>
        <Mnemonic>ChretienEUR</Mnemonic>
        <Currency>EUR</Currency>
        <CurrencyMarket>3</CurrencyMarket>
        <OpeningDate>1999-02-20</OpeningDate>
        <AccountOfficer>Soraya Lagarias</AccountOfficer>
        <LastUpdate>2004-02-10T22:33:58</LastUpdate>
        <Balance>
                <OnlineActualBal>896882</OnlineActualBal>
                <OnlineClearedBal>337676</OnlineClearedBal>
                <WorkingBalance>430147</WorkingBalance>
        </Balance>
        <Passbook>Yes</Passbook>
        <gValueDate>
                <mValueDate>
                        <ValueDate>1996-09-13</ValueDate>
                        <CreditMovement>6286.27</CreditMovement>
                        <ValueDatedBal>3065</ValueDatedBal>
                </mValueDate>
                <mValueDate>
                        <ValueDate>1996-12-09</ValueDate>
                        <CreditMovement>42300.08</CreditMovement>
                        <ValueDatedBal>86822</ValueDatedBal>
                </mValueDate>
        </gValueDate>
        <ChargeCcy>EUR</ChargeCcy>
        <InterestCcy>EUR</InterestCcy>
        <AllowNetting>Yes</AllowNetting>
        <gInputter>
                <Inputter>Soraya Lagarias</Inputter>
                <Inputter>CongDuc Bottreau</Inputter>
```

```
        </gInputter>
        <Holdings>
                <Position>
                        <Symbol>ZION</Symbol>
                        <Name>Zions Bancorporation</Name>
                        <Type>Stock</Type>
                        <Quantity>1927.719</Quantity>
                </Position>
                <Position>
                        <Symbol>ASEPX</Symbol>
                        <Name>AmSouth Select Equity I</Name>
                        <Type>Stock Fund</Type>
                        <Quantity>1177.619</Quantity>
                </Position>
        </Holdings>
</Account>
```