



Experts in Test Data

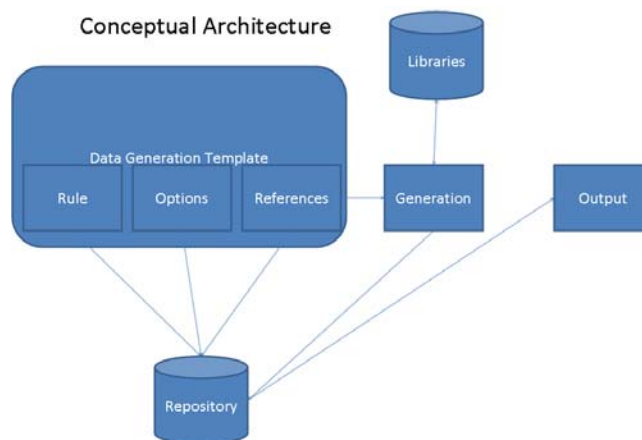
Many information technology applications such as Taxation and Finance, Healthcare, Intelligence or Biometric Security exist in an environment with legally or contractually mandated levels of privacy and confidentiality.

ExactData specializes in automating the generation of large sets of synthetic reference test data that meet the consistency and realism requirements that application performance testing demands.

Automated Data Generation for Testing or Training: Technical Capabilities Briefing

Capabilities Summary

Our unique patent pending technology is the automatic creation of multi-layered complex data record sets through a GUI to apply contextual rules, statistics, field and group dependencies, field and group dependencies upon dependencies, etc. Every element generated can establish a new context for subsequent processing in a data record set. Tables of data are established with constraints, boundaries, generation rules (sequential, random) of data on column basis and reference databases (libraries) for use to pull data.



Value

- **Lower Costs / Better Testing**
 - Reduce cost over manual data record creation processes with enhanced capabilities.
 - Flexibility to easily change, immediately creating new test or training case scenarios.
 - Output on demand in any file format (including paper or image).
 - Better, more thorough testing reduces development cycle times and drives higher quality.
- **Fake (Synthetic) Data Sets**
 - Completely eliminates security and privacy risks.
- **Large and Realistic Data Sets**
 - Representative, contextually correct, and statistically valid.
 - Deep system penetration (the data is correct to a deep level of analysis).
 - Robustly variable (because the tool is a tireless bookkeeper and assures broad and thorough coverage over the record data space).
- **Engineered Data**
 - Tailored from enterprise to component level testing for all of your enterprise solution testing requirements.
 - Large realistic data record sets with introduced anomalies that are flagged, creating the ability to test or train with a known response.
- **Perfectly Known TRUTH with Metadata**
 - Having the metadata that describes the usage and placement information of the *needles* in the data *haystack* is essential for proper system analysis and testing.
 - Default and custom metadata 'flags' enable a complete statistical analysis of the data set being used for testing.
 - Using the TRUTH and metadata together allows for full root cause analysis of a system to determine weak points, break points or bugs within a system.

Dynamic Data Generator™

Dataset Characteristics

Large Scale

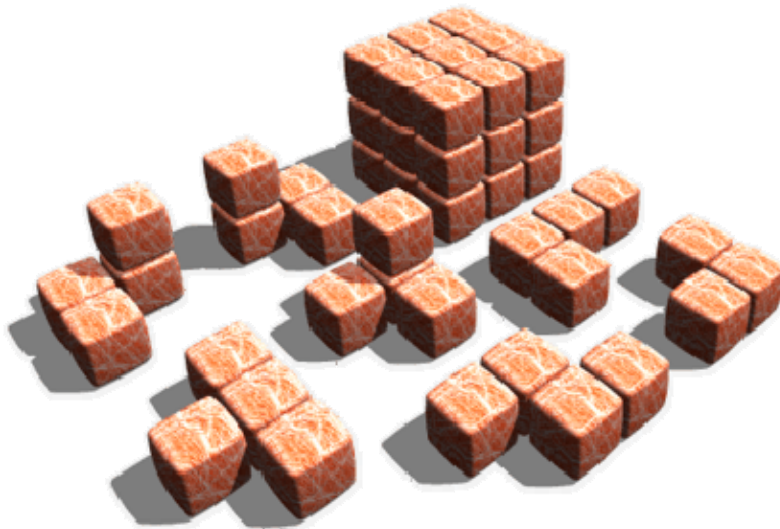
Synthetic

Realistic

Engineered

Wide variety of data types

The Dynamic Data Generator comes equipped with over 30 data generation controls. These include simple, generic controls (e.g. Random date generation), as well as complex industry-specific controls (e.g. modeling a household for the US 2010 Census). New data generation features & controls are being added regularly. Modeling data for your application can be likened to building using toy blocks – pick the blocks that make sense & combine them in a meaningful order.



-
- Generic data generation controls
 - String manipulation controls (concatenation, substring, injection, padding, formatting)
 - Random string chosen from a custom list
 - Random strings conforming to an Alphanumeric pattern
 - Numeric controls (random number in a static or dynamic range, mathematic manipulation of values, auto-incrementing)
 - Random date within

- Synthetic people & residential households
 - Name, gender, age, DOB, SSN, race, ethnicity, job
 - Various household structures (nuclear family, extended family, roommates, boarders, etc).
 - Phone, Address
- Industry-specific Models
 - Medical patients visiting emergency clinic
 - Biometric data models
 - Residential household in the US as viewed by the IRS
 - Businesses in the US as viewed by the IRS
- Complex conditional logic statements
 - Ability to specify complex causal relationships between data elements
 - Ability to combine logical operators and various state tests
- Extensively Configurable
 - Each widget in the data generation toolbox can be configured to allow precise engineering control of the output, e.g. average size of households, or Infectious disease outbreak affliction rates for 2-5 year olds.

Realism through Internal Consistency

The Dynamic Data Generator doesn't just produce *random* data; it is capable of producing *realistic* data that is internally consistent. The internal data integrity is achieved through multiple layers of consistency rules. Figure 1 illustrates this idea by showing the relationship between name data, personal identifying data, and household demographic data.

Layers of Consistency

Realism through internal consistency rules applied across many data elements

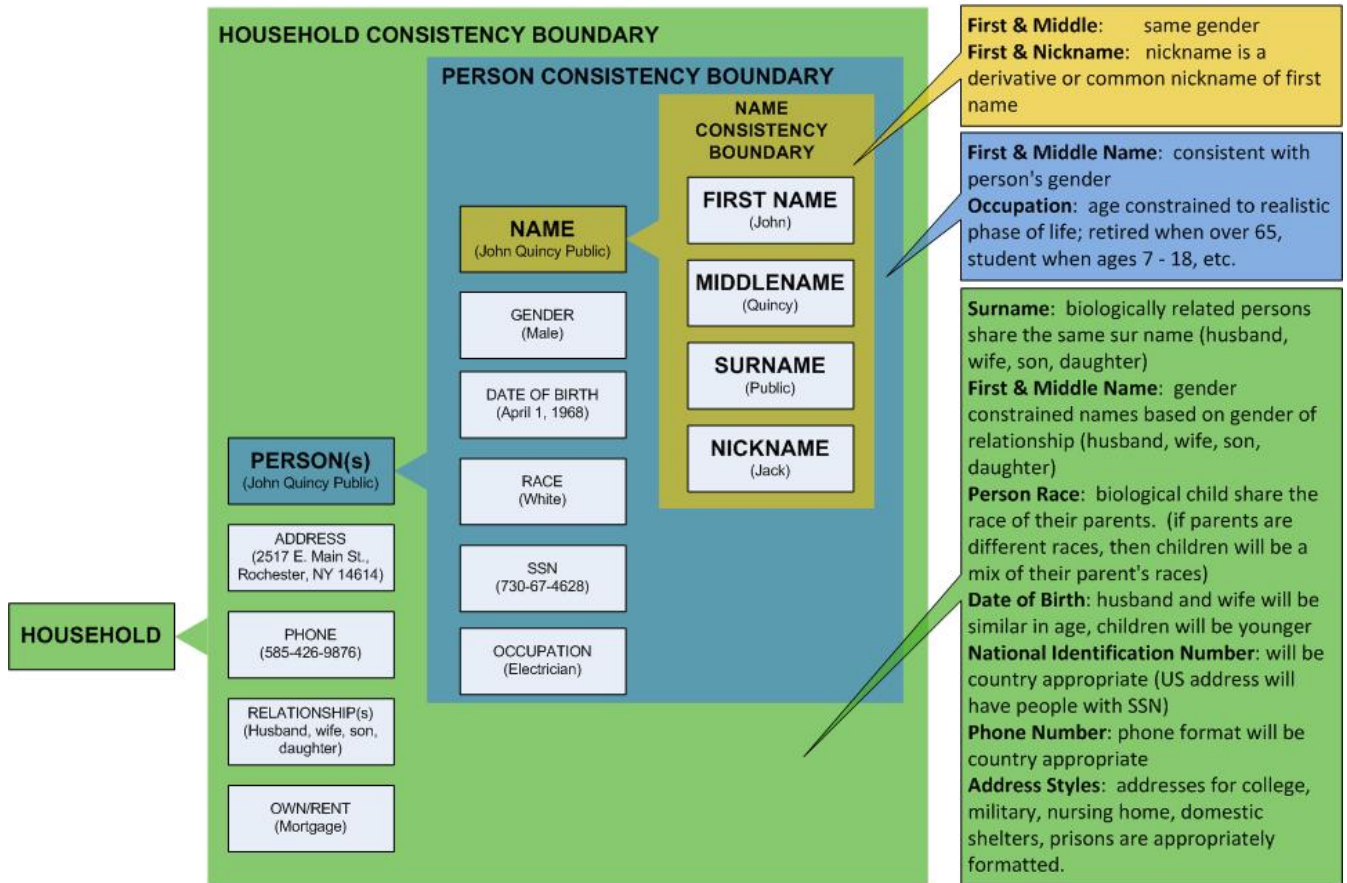


Figure 1: Internal Consistency

- Data for random names (Figure 1, yellow block)

The first name and the middle name are of the same gender. The nickname, if present, will be appropriate for the first name.

- Data for random people (Figure 1, blue block)

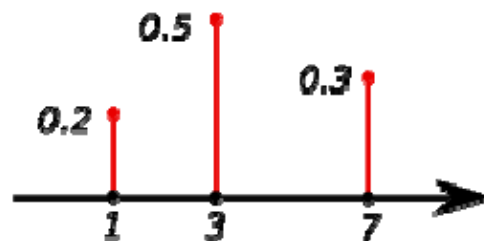
Gender consistency is maintained when Name is applied to a Person, such that the gender of the person is consistent with the gender of the first and the middle names. Person has other attributes including Date of Birth and Occupation. The occupation chosen for each person is constrained by choices that are sensible for the age of the person.

- Data for random households (Figure 1, green block).

New layers of consistency are added when a Person is added to a Household. Surnames of husband, wife, son and unmarried daughter are shared. Relationships such as husband, wife, son and daughter are also gender constrained. Biological children are a mix of the race of their parents. Children are generationally younger than their parents; this is reflected in their respective dates of birth. The household address must have a consistent format with the county where the household resides. Telephone numbers likewise are formatted according to country of residence.

This illustrates just one area of internally-consistent data generation. The Dynamic Data Generator is capable of an unlimited number of internally-consistent data sets using the rules which are meaningful for your application.

Controlled distributions of random variables



The Dynamic Data Generator allows control of dispersion patterns within randomly generated data. A simple example is that of the number of people within a residential household. If household sizes range from 1 to 12 people, you probably don't want a simple random distribution where there are as many 12 person households as 4 person households. Distributions can be defined according to shapes (such as triangular, bell curve, half bell curve, accelerating increase, semi circle) or by specifying the probability mass function (e.g., 20% of households generated having 1 person in them, 50% of households generated having 3 people in them, and 30% of households generated having 7 people in them).

Longitudinal Data Model

lon·gi·tu·di·nal (lɒn'jɪ-tɔːd'nəl, -tyɔːd'-, lɒn'-)
adj.

1. a. Concerned with the development of persons or groups over time: *a longitudinal study of twins.*

The Dynamic Data Generator supports the generation of “time-varying” or “longitudinal” data. Thus, you can generate data in which 1 or more records are longitudinally related to other records in the dataset. Each of these “time-varied” records is consistent with one another, but each contains some modified data

elements, in a manner one would expect from natural changes in the model as it varies over time.

For example, consider a fictitious medical patient, Mary White. Figure 2 illustrates various points in her life in which she sought care from a healthcare provider.

Time-varying Data

Longitudinality through internal consistency rules applied across time

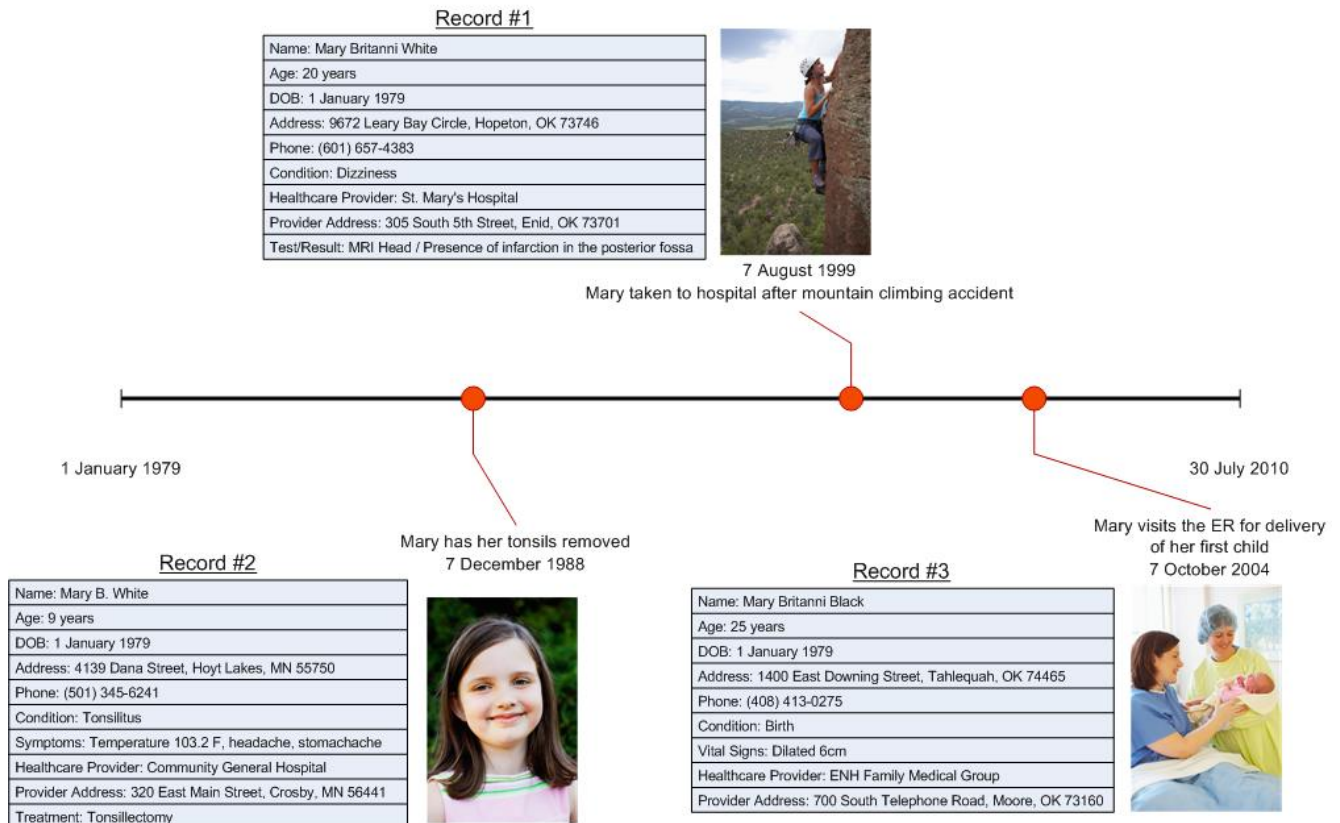


Figure 2: Longitudinal Data

- **Record #1:** The current date is 7-August-1999. At age 20, Mary White is treated for head injuries resulting from a mountain climbing accident. The data associated with this record includes her name, home address, healthcare provider visited, and results of medical tests performed.
- **Record #2:** The next record continues to model the same fictitious person, but with the data changed to be relevant to Mary's life at a different point in time. The current date is 7-Dec-1988, and Mary is 9 years old. She is visiting Community General Hospital to undergo a tonsillectomy. Though the data between the two records is consistent, differences between them are in evidence based on the different times in the two events (e.g., she lived at a different address at age 9 than

she did at age 20). There is also a difference in the specific text used for her name (“Mary B White” vs. “Mary Brittani White”) to indicate a natural variation in record-keeping that would likely occur in these two disparate events.

- Record #3: The last record indicates a childbirth event. The current date is 30-July-2010, but new changes are now seen in our patient’s actual name as a result of a changed marital status (“Black” vs. “White”) and again in her address, which differs from both of the two previous entries.

Although not shown in Figure 2, Mary's accident in 1999 continued to manifest in her life in the form of severe & debilitating migraine headaches for the next 10 years. Additional longitudinal records would have shown Mary performing lab tests (October 1994, elevated protein C discovered) and obtaining pain treatments (November 1994, 900mg Gabapentin, 3x/day).

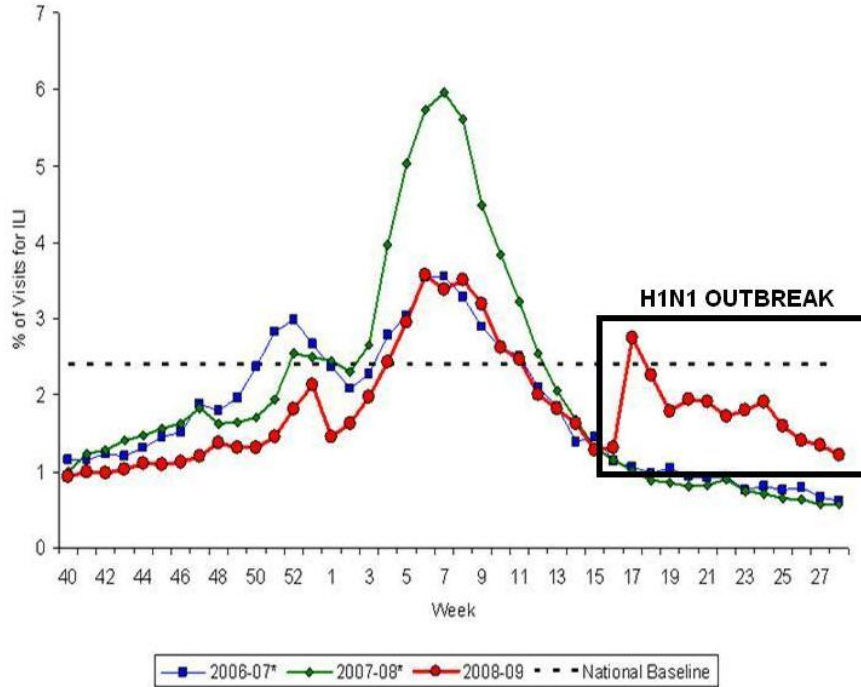
Pattern Injection

The Dynamic Data Generator has powerful pattern injection capabilities. It can generate complex data patterns, like the affliction pattern of a chronic illness upon a medical patient across time. It can also generate data that contains anomalous patterns - the proverbial needle in the haystack - that test the complex pattern recognition capabilities of your application.

Figure 3 shows Influenza Like Illness (ILI) incidents across 3 years. The illustration depicts ILI incidents compared as a percentage of total outpatient visits over time. Outbreaks of ILI are normal. While the green curve showed a higher incidence of ILI in the 07-08 season, it was the unseasonal spike in ILI in week 16 of 2009 that indicated that something new was happening. Using a needle & haystack analogy, one can see that the seasonal ILI outpatient visits becomes the haystack and the H1N1 outbreak is the needle. The DDG created a data set modeling this curve to enable testing of an Adverse Event Management system under development by one of our clients.

Percentage of Visits for Influenza-like Illness (ILI) Reported by the U.S. Outpatient Influenza-like Illness Surveillance Network (ILINet), National Summary 2008-2009 and Previous Two Seasons

(Posted July 24, 2009, 2:00 PM ET, for Week Ending July 18, 2009)



*There was no week 53 during the 2006-07 and 2007-08 seasons, therefore the week 53 data point for those seasons is an average of weeks 52 and 1.

Figure 3: Outbreak Pattern

Our tools enable the creation many types of patterns from simple (e.g. engineered errors like missing data) to complex (haystack/needle patterns). Knowing the answers in an engineered data set enables our clients to test and quantify their application's performance in finding ALL the right data and not producing false positives.

The following examples list some of the many applications of this pattern injection capability:

- Credit card fraud detection, e.g. online credit card orders with a few of the credit card numbers listed multiple times with different CVC # (security code on the back of the card).
- Terrorist detection, e.g. finding a person of interest through cross referencing checking account statement, cell phone bill, and databases of terrorist suspects.
- Engineered errors, such as a math error when totaling invoice line items or calculating tax.

- Increased service calls with specific complaints such as autos of a particular make having increased incidence of sudden acceleration.
- Localized voting patterns that may indicate fraud.
- Accident rates in some areas higher than average for the traffic volume.
- Specific chemical or prescription drug purchase skyrocket in some locations.
- Natural gas usage per distribution network can indicate a natural gas leak.
- Random localized cell phone outages indicating potential bandwidth interference.

Consistency across disparate data stores

The Dynamic Data Generator has the ability to generate data to create distinct stores of data that can be used separately, each containing unique data, but with critical data elements consistent among them. This provides a client the ability to use the same synthetic, dynamically generated data among a number of different applications and for a number of purposes.

In the figure 4 below, Dynamic Data Generator is used to produce a set of data stores that can be used to provide three different record sets. The checking account records describe a transaction by which money was deposited from a different checking account (circled in green). A cell phone bill shows the same person (name and address is common between the two) describes a series of phone calls. A third record set, a terrorist suspect watch list, contains data from both the checking account statement and the cell phone bill. These three disparate data records, each containing unique as well as common data elements, come together to tell a story would be of interest in a security data mining application- a known terrorist was both wiring money and talking to a person within the United States.

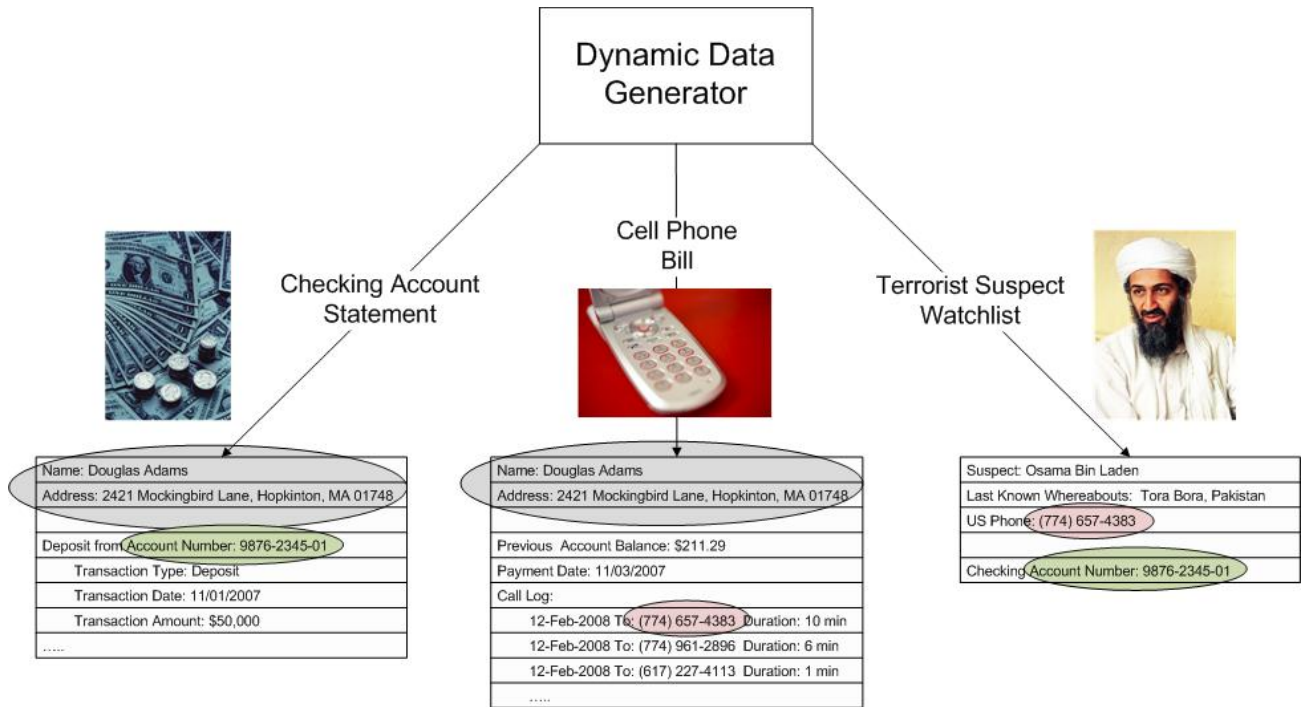


Figure 4: Disparate Data Stores

The Dynamic Data Generator can format and export data in industry-standard formats, such as CSV and HTML, as well as in application-specific formats, based on XML or according to custom requirements from specific applications.

Unstructured data generation

In addition to generating structured data records, the Dynamic Data Generator has the ability to generate meaningful unstructured synthetic datasets. Clinical notes is an example of our current capability. For a record of data representing the issuance of billable healthcare services rendered to a medical patient, the clinical notes authored by the attending physician need to be consistent with the structured data in the record set (Figure 5).

Date:	6/26/2008
Patient:	Cody B Marsh
Patient_DOB:	9/30/1924
Patient_Address:	RR 47 BOX A6
Patient_City:	Jamesville
Patient_State:	VA
Patient_Zip:	23398
Patient_Height:	175 cm
Patient_Pulse:	61/min
Patient_Temp:	36.7 Cel
Patient_Weight:	122 kg
Provider_Name:	Meadows Medical
Provider_POC:	Nicole Diaz, MD
Provider_Address:	101 Elm Avenue
Provider_City:	Roanoke
Provider_State:	VA
Provider_Zip:	24013
Condition:	Migraine Headaches
Drug_Name:	Amitriptyline
Drug_Dosage:	50 mg; daily @ bedtime
Notes:	s:84 year old male presents with Migraine-like symptoms. o:Pulse 61/min, Temperature 36.7 Cel, Height 175cm,Weight 122kg a:Migraine Headaches p:amitriptyline 50 mg; daily @ bedtime

Figure 5: Clinical Record

Other types of free-text or unstructured data generation is also possible:

- Emails
- Poems
- User Manuals
- Whitepapers

The Generated TRUTH File and Metadata Within

Each data set(s) produced by the DDG includes a TRUTH file containing metadata. Efficient and complete testing requires complete knowledge in the data and metadata contained by test object. This level of confidence is essential of both the test object and the desired resting results is essential to be truly be effective in testing; system testing, performance testing, etc.

Knowing a data set contains a desired anomaly is good. But knowing where, how often and in what manner a certain desired anomaly appears is even better, and is where the metadata within the TRUTH files comes into play. This metadata provides the characteristics of the anomaly including where the anomaly appears.

The TRUTH file not only provides the expected value for a certain field, the metadata within the TRUTH file contains additional information such as the type and value of anomaly, as well as the way in which it was introduced. This information is essential to root cause analysis on a system's reaction to a certain set of data, or particular value. The TRUTH file along with its metadata provides a complete characterization of the input test object.

Metadata can not only characterize desired statistical anomalies when they occur but can also be used to highlight when certain business or process rules have been purposefully broken within the data. System testing much like black box testing is often performed without knowledge of a systems interworkings. The addition of the descriptive metadata to the TRUTH file, the system can now be characterized using the 'why' behind each failing anomaly. Thusly, the combination of ExactData's data set(s), TRUTH file and its included metadata make for a superior and more complete testing object. The combination of which enables the ability to test the manifestation of nearly any stress, performance, or load test requirement while being able to provide thorough root cause analysis for any system or subsystem failure(s).