



TEST DATA GENERATION USE CASES

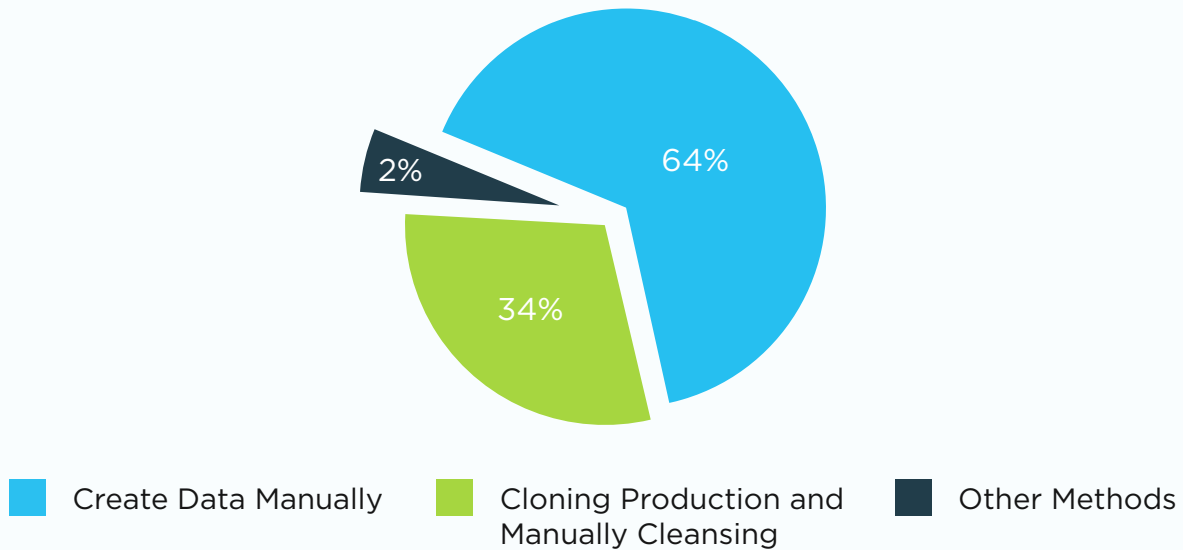
10 Ways to Use Synthetic Test Data to Increase Efficiency



Quality assurance teams historically struggle with provisioning test data for their automated test environment. A survey of DevOps professionals (Delphix: 2017 State of Test Data Management) describes test data provisioning as a “slow, manual and high touch process” reporting that 3.5 days and 3.8 people were needed to fulfill the average request for test data. For 20% of DevOps professionals, the provisioning timeframe was over a week. Test data has become a bottleneck and a barrier, especially for continuous integration and delivery environments.

GenRocket’s synthetic Test Data Generation (TDG) platform is designed to eliminate the bottleneck by creating test data for virtually any test case on-demand and in real-time.

AVERAGE TIME SPENT ON TEST DATA MANAGEMENT (TDM)



A recent study by IBM found that 30% to 60% of a testers time is spent searching, managing, maintaining, and generating test data. This puts QA managers in a difficult spot as they look for ways to streamline the provisioning process and utilize their testers more efficiently. GenRocket customers have learned how to leverage the power of TDG for any type of testing operation (e.g., unit, functional, integration, performance, regression etc.) to accelerate and improve the quality of test operations.

Most GenRocket customers begin their journey into the world of TDG with a Proof of Concept (POC) to experiment with the technology and validate the use of real-time synthetic test data in their automated test environment. Once they successfully complete the POC, they are ready to broadly deploy GenRocket into their test environment. However, sometimes even the most experienced QA managers need guidance for how to introduce TDG into their organization.

After a POC, the QA manager's attention changes from asking *"should I be using synthetic test data to replace production data"* to the more practical question: *"when, where and how should I use synthetic test data for maximum impact."*

We developed the following guidelines to help our customers follow the most effective path for TDG deployment. Our goal is the same as their goal: Introduce GenRocket one application at a time in a way that maximizes the impact on the quality and efficiency of test operations. The following 10 questions will help you identify some "quick wins" for transitioning to GenRocket as you gradually increase the level of synthetic test data used for software testing from 5-10% to a level of 90-95% of your total test data requirements. In general, our customers find the higher the percentage of real-time synthetic test data that is generated (versus production data), the greater the level of efficiency and quality achieved throughout the QA operation.

As you assess the applications environment, data environment and test environment in your organization, ask yourself these questions to identify the most impactful use of GenRocket.



1. DATA PRIVACY:

Where does your test data contain sensitive customer information?

Your company may have a mandate to remove Personally Identifiable Information (PII) from test data to avoid the risk of a data breach and/or ensure compliance with privacy regulations. Even if you don't, GenRocket can instantly provide synthetic test data to take the place of pruned, masked and obfuscated production data. Eliminating PII from the testing process is one of the first areas where GenRocket can have an immediate impact.



2. BOUNDARY CONDITIONS:

Which of your test cases require boundary condition testing?

Test cases often call for testing boundary conditions, which means testing all edge cases to identify unexpected outcomes. This requires test data that has been conditioned for minimum/maximum values, a wide variety of data patterns and the use of all potential data permutations. GenRocket has 213 data generators with specific Generators for boundary conditions and permutations and Generators for all common test data categories (see below).

Pattern	Realistic	Sequential	Random	Edge Case	Null
firstName1	Ms. Tereasa F. Saldana	001-01-0001	749-40-0182	749-40-0182	749-40-0182
firstName2	Mr. Everette Q. Groom II	001-01-0002	797-59-7445	797-59-7445	null
firstName3	Mr. Jules U. Hackney Jr.	001-01-0003	135-93-8060	135-93-8060	135-93-8060
firstName4	Mrs. Kristina J. Brick	001-01-0004	214-82-8447	214*82*8447	null
firstName5	Mr. Francisco M. Grimes II	001-01-0005	170-60-5224	170-60-5224	null
firstName6	Dr. Iona D. Starrett	001-01-0006	302-76-0978	302-76-0978	null
firstName7	Ms. Patricia O. Ingraham III	001-01-0007	266-20-5659	266-20-5659	266-20-5659
firstName8	Ms. Tracee M. Farah	001-01-0008	005-57-7667	005#57#7667	005-57-7667
firstName9	Mr. Alva I. Ziegler Jr.	001-01-0009	490-48-8084	490-48-8084	null
firstName10	Dr. Mike T. Youngblood II	001-01-0010	471-29-7519	471-29-7519	null



3. BLENDED TEST DATA:

Do you want to blend production and synthetic data?

Sometimes you want to leverage specific sets of program data in your tests and combine that data with secure, controlled, synthetically generated test data. For example, GenRocket can query program data like Account ID's or any other enumerated type of data and pull that data into a list as part of a test data generation scenario. In another use case GenRocket can use synthetically generated test data to create the conditions to retrieve a specific program data value. For example, GenRocket might synthetically generate a male gender, of age 50 and an income greater than \$60,000 and those conditions would be used to query and retrieve a specific value from the production database environment. This shows a powerful solution where synthetic test data is blended with production data.



4. VOLUME DATA:

Which of your tests require the highest volume of test data?

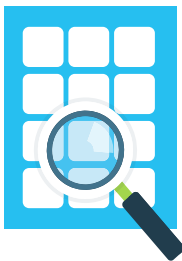
If your test operation requires high volume test data to conduct application performance testing or endurance testing, GenRocket has the ability to generate synthetic test data at the rate of 10 thousand rows per second. And through the use of the GenRocket Partition Engine, a billion rows of test data can be generated in under 20 minutes. This level of scalability also positions GenRocket as a test data source for big data and machine learning environments.



5. DATA REFRESH:

What test data is subject to frequent changes and data refresh?

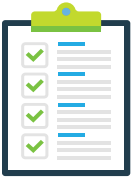
When test data needs to be frequently refreshed, GenRocket can save time and streamline the provisioning process by generating test data sets that reflect the most current data model. Testers can refresh data sets on-demand, in real-time simply by executing GenRocket Scenarios that reflect the current data model.



6. PAIRWISE TESTING:

Can you get full coverage with only a fraction of the tests?

How do you get much better test coverage without having to run hundreds or thousands of tests? All-pairs testing or pairwise testing is a combinatorial method of software testing that, for each pair of input parameters to a system, tests all possible discrete combinations of those parameters. The Pairwise approach is really compelling for testers but historically the challenge has been the complexity and how to provision the right test data for the Pairwise tests. GenRocket is now offering a Pairwise testing solution that is easier to use and with integrated test data provisioning.



7. SPECIAL FORMATTING:

Does application testing require data with special formatting?

Depending on the application under test, your data may need to be in a special format like XML, JSON, CSV, DB2, VSAM, SQL or even an industry-specific format like HL7 for health care. GenRocket Receivers generate test data in over fifty different formats to meet the compatibility needs of any application. If you need a special test data format, consider GenRocket for meeting this requirement.



8. NEW APPLICATIONS:

Are you testing applications for which there is no production data?

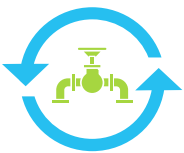
When new applications are in development, there may be no production data available to use for testing. Until now, the only alternative was to manually create the test data or wait until the application was released to production and then start testing. With GenRocket, synthetic test data designed for that application can be used for all types of tests including unit, integration, functional and load testing of the new application before it is released to production.



9. DYNAMIC TEST DATA:

Do you need dynamic test data for end to end / workflow testing?

Provisioning test data that changes dynamically in real time is needed for effective end to end / workflow testing. A test data generation system with a powerful API is a must for end to end testing as there are so many combinations and rules that impact the test data. GenRocket's API can be programmed to look at the result of a test or query and dynamically change the test data that is generated based on that result. A whole new level of automated end to end testing is possible with API-driven, dynamically generated test data.



10. CI/CD PIPELINES:

Which applications have tight deadlines and frequent revisions?

In a continuous testing environment, test data must also be continuously available. The GenRocket TDG platform was built for speed and versatility to accelerate any form of testing with data that reflects the most current version of the data model. Integrate GenRocket into a CI/CD Pipeline with Jenkins for applications that have high frequency revisions and rapid time-to-market requirements.

DEVELOP YOUR OWN TEST DATA GENERATION DEPLOYMENT PLAN

A summary of these guidelines for *Test Data Generation Deployment* is provided in the checklist below. Use it to assess your applications and their respective test operations to target your own priorities for introducing synthetic test data generation into your QA environment.

Test Data Generation Deployment Checklist			
TDG Guidelines	Applications	Test Operations	Data Source
1. Data Privacy	<i>Application Name</i>	<i>Test Cases</i>	<input type="checkbox"/> TDG
2. Boundary Conditions	<i>Application Name</i>	<i>Test Cases</i>	<input type="checkbox"/> TDG
3. Blended Test Data	<i>Application Name</i>	<i>Test Cases</i>	<input type="checkbox"/> TDG
4. Volume Data	<i>Application Name</i>	<i>Test Cases</i>	<input type="checkbox"/> TDG
5. Data Refresh	<i>Application Name</i>	<i>Test Cases</i>	<input type="checkbox"/> TDG
6. Pairwise Testing	<i>Application Name</i>	<i>Test Cases</i>	<input type="checkbox"/> TDG
7. Special Data Formats	<i>Application Name</i>	<i>Test Cases</i>	<input type="checkbox"/> TDG
8. New Applications	<i>Application Name</i>	<i>Test Cases</i>	<input type="checkbox"/> TDG
9. Dynamic Test Data	<i>Application Name</i>	<i>Test Cases</i>	<input type="checkbox"/> TDG
10. CI/CD Pipelines	<i>Application Name</i>	<i>Test Cases</i>	<input type="checkbox"/> TDG

RECOMMENDED TEST DATA GENERATION DEPLOYMENT MODEL

The checklist can be used as part of a formal process to deploy TDG across the organization. After initial deployment, the use of TDG should be increased methodically, one application at a time, to expand the percentage of synthetic test data to 90% or more of all test data. This will maximize the efficiencies realized in terms of provisioning speed, cost-effectiveness, data quality and overall simplicity of user operations.

Here are the steps we recommend for organizations getting started with TDG:

1. Identify up to three GenRocket TDG use cases. Conduct POC's (*Proof of Concepts*)
2. Purchase licenses and deploy GenRocket for the POC use cases.
3. Conduct an assessment of the application environments, data environments and testing environments. Develop a prioritized TDG deployment plan.
4. Expand the use of GenRocket based on the prioritized TDG deployment plan.
5. Measure synthetic test data use against a target of over 90% synthetic in all test data use cases

The following framework provides a structured approach for conducting your assessment of the application environments, data environments and testing environments.

APPLICATION ENVIRONMENT

Understand the applications and infrastructure that supports them and underlying databases.

Deliverable > *Applications Audit*: Identifies applications and infrastructure.

DATA ENVIRONMENT

Understand the information flow through the application environment and the data.

Deliverable > *Data Audit*: Identifies data flow and data dependencies.

TESTING ENVIRONMENT

The current testing resources, test processes, testing tools and data provisioning processes.

Deliverable > *Test Audit*: Test data tool and framework integrations.

TDG DEPLOYMENT

The TDG deployment plan for the organization.

Deliverable > *TDG Deployment Plan*: Prioritized plan for application testing, revised processes for test data provisioning, data refresh and test data integration with tools and frameworks.



Do you need assistance with TDG deployment? GenRocket can help by coordinating a professional services engagement with one of our partners to design and deliver a Test Data Generation roadmap tailored to meet the specific needs of your QA organization. If you would like to know more about TDG deployment assistance, please email info@genrocket.com.