PARASOFT®

WHITEPAPER

The Solution to Your **Test Data Management** Headaches

TIRED OF WAITING FOR TEST DATA?

Test data management (TDM) is a big problem for organizations, with the process of procuring, owning, and securing test data both a requirement and a liability. Good test data is hard to come by, and test teams are often left waiting for test data to be produced for them.

TDM is a requirement because without proper test data it's difficult to achieve a high level of test coverage, but TDM is a liability because test data contains sensitive information that could be misused and leaked beyond the organization, presenting financial and legal implications and risk.

Parasoft provides a modernized solution to solve test data management headaches. Parasoft's data simulation approach combines traditional test data extraction along with service virtualization, in an easy-to-use web interface.

Users can quickly build meaningful test data by capturing realistic test data from interactions between components in their existing system (real and virtualized), and building data models that can be shared and controlled directly by testing teams.

TRADITIONAL APPROACHES TO TEST DATA COLLECTION

There are many options for procuring test data. The first and most obvious is to collect data from a running production system, literally cloning the production database. This data has the right content and characteristics but is the least secure and can contain sensitive customer information that could be compromised.

Another option is a partial, subset of the production database. This approach means less data to manage but does suffer from the same privacy and security concerns as a full clone. Data synthesis is a possible alternative, but it's difficult to generate the data in the proper form without having domain knowledge. The required expertise may not be available.

Consider some troubling statistics. According to a recent survey from Sogeti, up to 60% of application development and testing time is devoted to data-related tasks, and many project overruns are due to inefficiencies in test data provisioning.

In general, up to 20% of the average software development lifecycle is lost waiting for data! Clearly, software teams need a maintainable process to procure an adequate amount of data while still keeping that data safe.

A TEST DATA MANAGEMENT SYSTEM THAT WORKS

Let’s consider what’s needed to make test data management less of a cost and risk in software testing. Ideally testers shouldn’t be held up waiting for test data but they shouldn’t also be saddled with data that has security and privacy issues embedded in it. Also, they need a test automation solution that’s not just running and monitoring their tests, they need management of the data as well. Consider the following requirements for test data management that solves issues facing testers today.

REDUCE TIME SPENT WAITING FOR DATA

Testers can quickly create test data by capturing, masking, and repurposing realistic test data from interactions between real and virtualized components in their existing systems. Using this captured data it’s possible to build data models that are shared and controlled directly by the testing teams.

REDUCE THE COMPLEXITY OF DATA STEWARDSHIP

There must be a maintainable way to procure an adequate amount of data while still keeping that data safe. Intelligent masking of any real data used is essential.

EFFICIENT ACCESS TO TEST DATA

Instead of getting data from a centralized test data management system, testers must be able to access, manage, model, and generate just the right data for their needs. Self service for each tester is essential.

EASIER TO UNDERSTAND TEST DATA

Traditionally captured live test data requires testers to have intimate knowledge of how the actual databases work (domain knowledge, database schema, and connection information). A better approach is to help infer what the actual data looks like from the traffic, reducing the learning curve required to take advantage of the test data.

INTUITIVE, VISUAL WAYS TO UNDERSTAND DATA SOURCES

Data sources are typically represented as flat files (CSV, XLS), however, it’s easier to understand in a hierarchical, visual way. Complex data objects that have parent-child relationships, visualized, are more intuitive and easy to restructure, so testers can reduce the time it takes to reconfigure test data with hierarchical relationships.

SIMPLIFYING TEST DATA MANAGEMENT WITH PARASOFT

Parasoft's self service web portal enables multiple team members to access, manage, model, and generate the right data for their needs. But data can't provide significant value to the organization unless there's an easy way to use that data. Parasoft's test data solution tightly couples with Parasoft SOAtest for functional test automation and Parasoft Virtualize for service virtualization, allowing users to take that generated data and use it in their tests, making it available through virtual interfaces, whether it's a REST service or a virtual database. Consider the following workflow diagram:

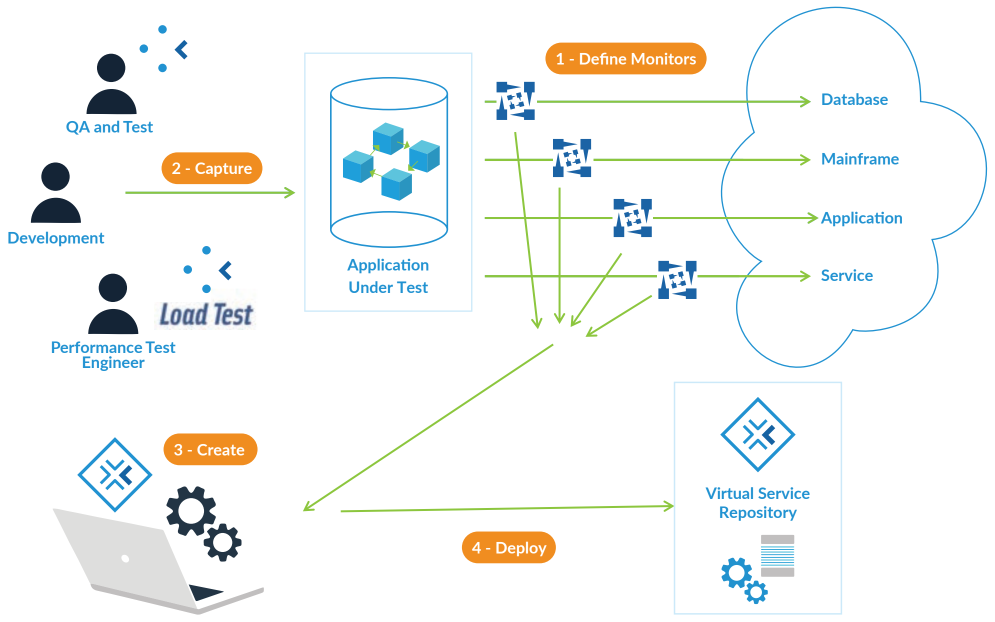


Figure 1:
Capture live traffic data to create test data models to use with virtual services.

HOW TO USE PARASOFT SOATEST AND VIRTUALIZE FOR TDM

1. **Define monitors.** Monitors are proxies that intercept and record the data traffic between users and servers.

2. **Capture.** Test data is captured by monitors and recorded from existing testing via functional, performance and security testing already in place by the development team. The application under test likely has dependencies on various types of services such as databases, legacy mainframe systems and many others. As test automation is executed the transactions and data between the application and services is recorded by the monitors and stored in the virtual service repository.

3. **Create.** The captured transactions and data are sent to the service virtualization engine to create simulated virtualized services with associated test data.

4. **Deploy.** The created services are stored and deployed to a virtual service repository allowing for later consumption.

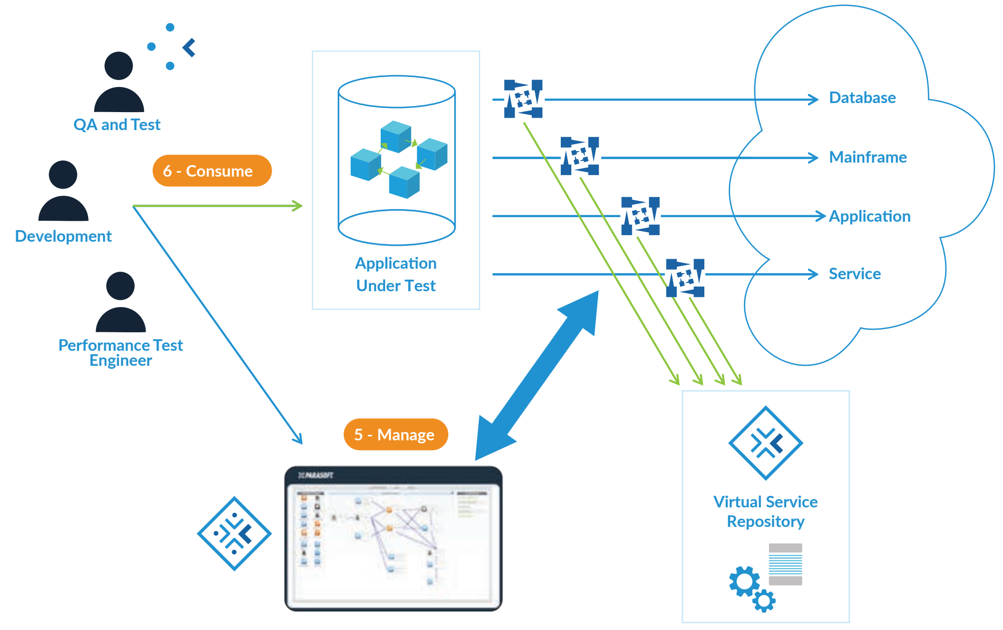
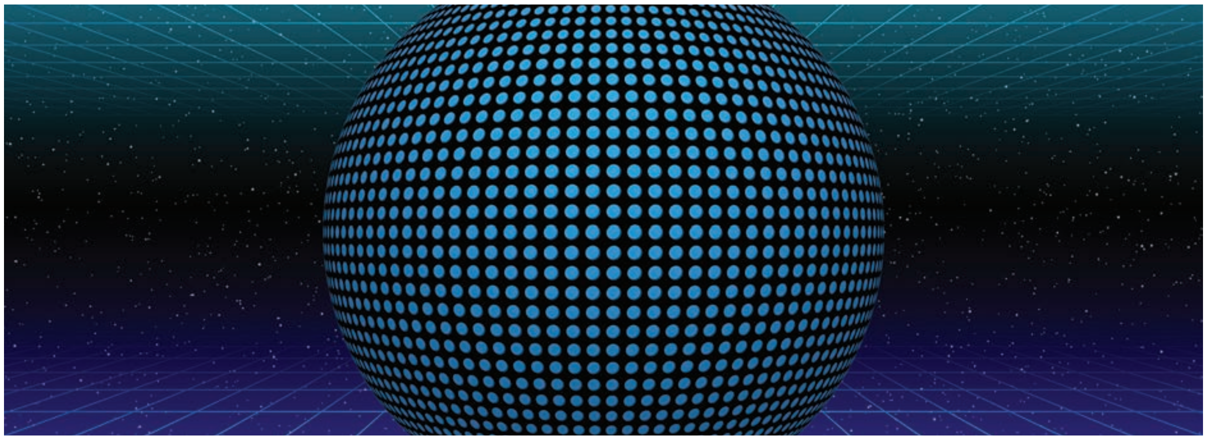


Figure 2:
With virtual services and test data defined, thorough testing can be performed for the application under test.

5. **Manage.** Orchestration of these virtualized services is required to make service virtualization work in a CI/CD workflow.
6. **Consume.** Test data management and integrated orchestration of these virtual services means automated API and service-level testing is robust, repeatable, and simple.



TEST DATA MANAGEMENT WITH SERVICE VIRTUALIZATION

Parasoft's test data management technology is augmented with service virtualization. A key example is replacing a reliance on a shared database by swapping it with a virtual service. This allows for parallel and independent testing, that would otherwise conflict. Parasoft's test data management engine extends the power of service virtualization by allowing testers to generate, subset, mask, and create individual customized test data for their needs.

By replacing shared dependencies such as databases, service virtualization removes the needs for the infrastructure and complexity required to host the database environment. In turn, this means isolated test suites and the ability to cover extreme and corner cases. Although the virtualized dependencies are not the "real thing," some actions, for example an insert and update operation on a database, add some complexity to virtualization.

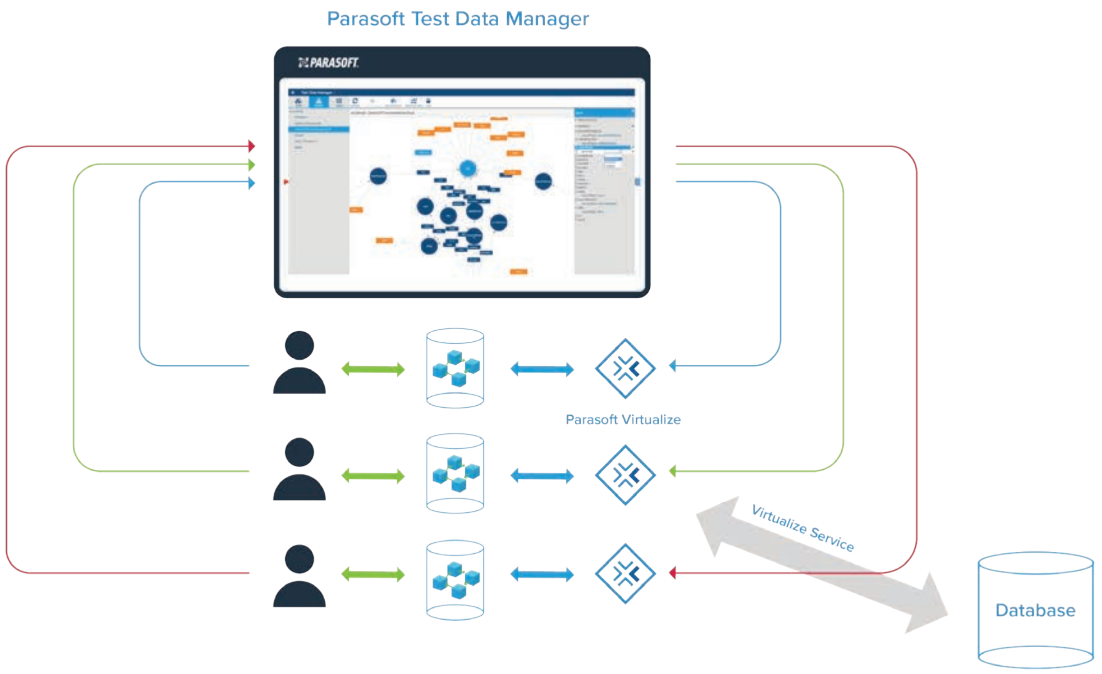


Figure 3:
The self service portal enables users to capture, mask, generate, and subset test data. Increase productivity and reduce resource constraints by testing with virtual data instead of the actual database.

After capturing transactions and data, there is full control over the contents of the data from within the data manager. In addition, a model is abstracted automatically from the data based on database extraction or through the interactions observed during the recordings. This model, as illustrated below, allows for better understanding of the data structure and relationships. This understanding is key to the next steps of masking, generating, and subsetting.

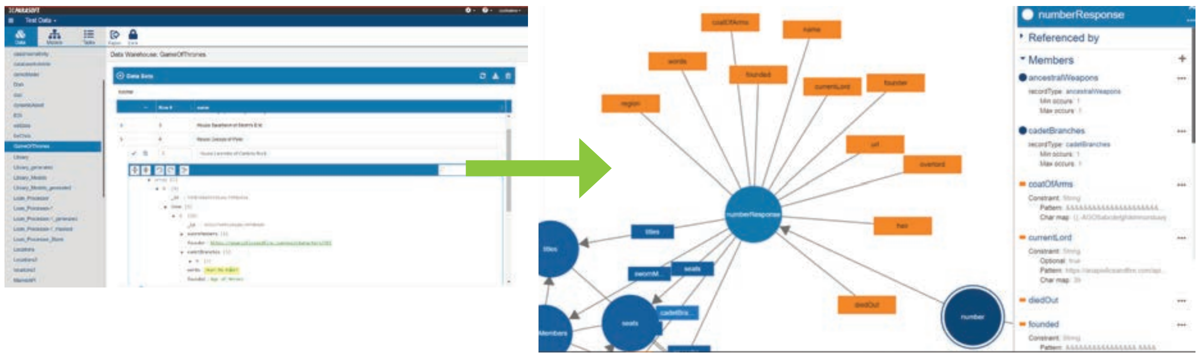


Figure 4:
Modify generated test data, and view the data structure and relationships within a model diagram.

A HOLISTIC APPROACH TO SERVICE VIRTUALIZATION & TEST DATA MANAGEMENT

Parasoft Virtualize and a test data management system like DatProf can be combined as well. Service virtualization is ideal when isolating the application from dependencies that restrict the flexibility of testing, and in cases where it's impractical, the traditional test data management solution makes sense for testing dependencies such as the application database. The combination of these two approaches is ideal for complementing the strengths of each approach.

TEST DATA LIFECYCLE

Once data has been captured, analyzed, and modeled, it's now possible to manipulate the test data to ensure reuse as well as privacy and security. This practice is known as masking. Parasoft's test data management technology supports a lifecycle approach for managing captured test data. This workflow follows these four main activities:

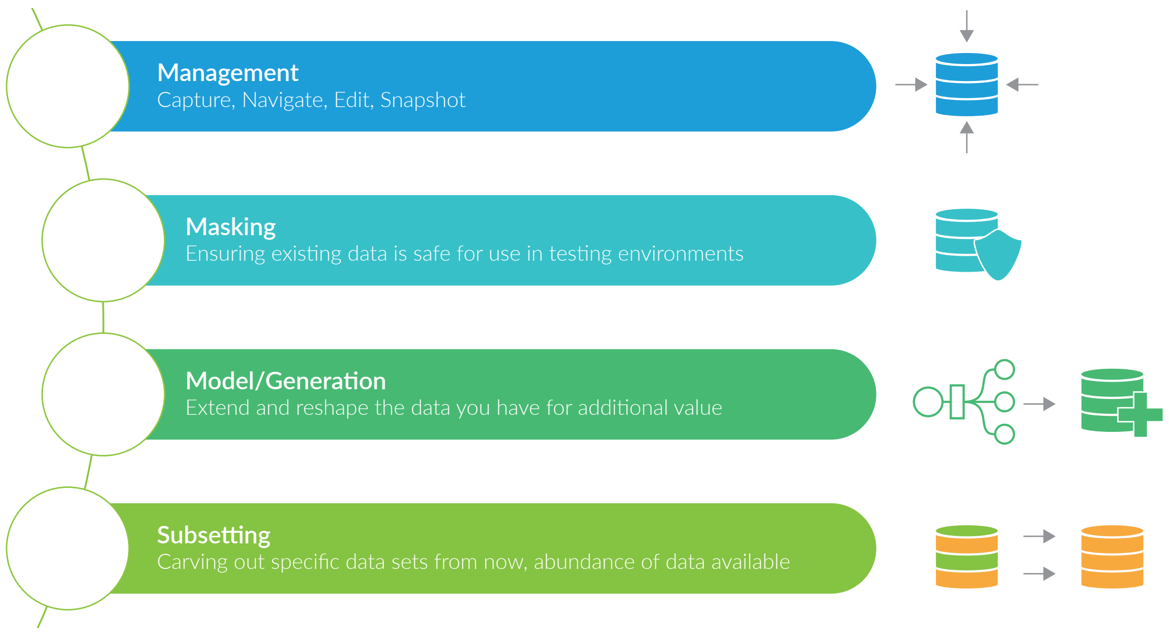

Management
Capture, Navigate, Edit, Snapshot
Masking
Ensuring existing data is safe for use in testing environments
Model/Generation
Extend and reshape the data you have for additional value
Subsetting
Carving out specific data sets from now, abundance of data available

Figure 5:
Work with test data using these four activities in the test data lifecycle approach.

MANAGEMENT

Capturing test data is done through monitors and recording transactions and data, or via database extraction directly. You must manage the datasets, models generated during recording, and manipulation of data. With Parasoft, all of this is simplified with a user-friendly, lightweight web interface to manage test data recordings, masking, subsetting, and generation.

MASKING

Test data can contain private and sensitive information, and in order to prevent security and privacy issues with the test data, the data is masked with different values. This is tedious to do without automation, and Parasoft's test data management solution makes this easy. Also, in cases where masked data forms an index (for example, social security numbers), referential integrity is maintained when these numbers are masked. An audit trail is generated by the tool when data is masked for privacy and security compliance.

Masking isn't simply replacing values with a constant letter or number, nor scrambling values. For instance, social security numbers would need to have realistic but not real values. For example, the SSN 354-15-1400 can't be replaced with XXX-XX-XXXX and be meaningful to the application. On the other hand, an intelligent masking operation replaces the real numbers with something that is recognizable by the application but not a real SSN, for example, 354-15-1400 becomes 004-15-1453.

There are other factors to consider when masking, and automation makes it much simpler, such as validity, format, and distribution of the data. Moreover, multiple copies of the same data must be masked identically, and referential integrity must be maintained.

MODEL/GENERATION

Recorded data is only so useful. The real benefit to recording and modeling is the ability to expand the data coverage of tests based on the recording. The model generated by Parasoft's test data management solution is useful in creating new test data that is in the correct format and content for the application, but with a wide range of customization. It is also possible to use seeds to generate new data or import external data to create a larger and diverse test dataset. Formulas can also be used to generate new data, which is a relationship of other data elements.

Once individual records are customized, generation of test data follows. Parasoft enables the tester to include original data in the generation, or the newly generated data, or a combination of both. The generated data source is then linked to a virtualized service, which then can be used for functional tests. This is an easy, automated approach to injecting useful test data based on real-world recordings.

SUBSETTING

The pursuit of test data for unique requirements is often a painful, manual effort. With Parasoft Virtualize and test data management, it's simpler to generate effective test cases for these unique situations using a process called subsetting. Unlike modeling and generation, subsetting is more akin to filtering out data that doesn't fit certain constraints. These data subsets are based on a set of rules that are now powerful test cases when paired with a virtualized service.

TAKE ADVANTAGE OF THE PARASOFT TEST DATA MANAGEMENT SOLUTION

SIMPLIFY TEST DATA MANAGEMENT

Users can navigate, edit, and manipulate their data structures in Parasoft's thin-client interface to reduce the amount of time spent looking for the right data.

SHIFT-LEFT INTEGRATION TESTING

Service virtualization is used to share data between test tools and the service virtualization layer to fully test an application and not be constrained by back-end systems. Simplified data storage is used rather than full schemas, which speeds up prototyping for better agility.

SIMPLIFY TEST DATA STORAGE

Built on top of a lightweight data storage mechanism, Parasoft's test data management solution lets you share and access data from a remote repository, where it can be easily managed, manipulated, and used in test cases and virtual services.

MANAGE COMPLEX DATA

Parasoft enables users to alter the shape of complex hierarchical data (add/remove occurrences, exclude parts of data, and so on) without having to update the database schema or service definition.

EXTEND AND RESHAPE DATA WITHOUT INTIMATE DOMAIN KNOWLEDGE

Model data objects and generate new data quickly. Create data on the fly for your test cases and virtual services, and free up your reliance on a centralized data management team.

SECURE EXISTING DATA FOR USE IN TESTING

Data masking/obfuscation secures test data to make it usable in test environments, and helps users leverage production data by securing the data after recording.

EXTEND SERVICE VIRTUALIZATION WITH TEST DATA

Users can augment their existing service virtualization strategy with flexible test data that automatically builds meaningful models by simply creating virtual services.

CHOOSE JUST THE DATA YOU NEED

Leverage data subsetting to carve out specific data sets from newly abundant data available, reducing overall data storage required by selecting just the data that's required. Generate, subset, then destroy.

SUMMARY

Test data is hard to procure and a risk to manage. A method based on secure, real data capture provides the best solution for attaining the data. Service virtualization provides additional benefits in capturing this data earlier in the development process. Parasoft's modernized test data management solution provides secure storage and management of this test data while allowing testers without in-depth domain knowledge to customize the data sets for their needs. Simplifying test data management reduces the risk and liability of using production data plus reduces overall project risk and costs.

LEARN MORE

See how a modern, holistic TDM solution increases efficiency for development teams. Watch the on-demand demo now.

TAKE THE NEXT STEP

Find out how to get your team started on a TDM solution that's simple to use and understand, quick to get data, and evolves with your application. Request a demo.

ABOUT PARASOFT

Parasoft helps organizations continuously deliver quality software with its market-proven, integrated suite of automated software testing tools. Supporting the embedded, enterprise, and IoT markets, Parasoft's technologies reduce the time, effort, and cost of delivering secure, reliable, and compliant software by integrating everything from deep code analysis and unit testing to web UI and API testing, plus service virtualization and complete code coverage, into the delivery pipeline. Bringing all this together, Parasoft's award winning reporting and analytics dashboard delivers a centralized view of quality enabling organizations to deliver with confidence and succeed in today's most strategic ecosystems and development initiatives — security, safety-critical, Agile, DevOps, and continuous testing.