



Success Story

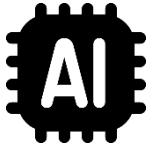
# Test Data Generation for an AI-Driven Platform

On-Demand Data  
Warehouses

Learn more at [curiositysoftware.ie](https://curiositysoftware.ie)

© Curiosity Software Ireland

# On-Demand Data Generation



A global 2000 software vendor had launched their flagship, AI-driven project management platform, but were struggling to create accurate test data.



They couldn't mirror large implementations in test environments, slowing bug fixes and damaging enterprise client relationships.



Today, they generate on demand data across a complex SQL Data Warehouse, while automatically documenting the system API and data model.

## Benefits at a Glance



Quality Engineering understands invalid data combinations for rigorous testing.



From creating 100s of rows of data manually, to generating millions on demand.



A gold copy data set that is 50% bigger than all client data combined.



18 synthetic databases drive parallelized testing and development.



Data warehouses reflect years' long, enterprise implementations.



An intuitive, visual approach to defining data and complex business logic.



High-performing generation mirrors enterprise software usage.



Generation of data across databases creates a complete Data Warehouse.



Functions and modelling reflect dependencies, historical logs, events, and temporal trends.



Automated data comparisons and validation document the underlying data model.



Auto-generated documentation details how APIs and databases function.



Developers understand the impact of their changes on back-end systems.

# Better Data, Faster Delivery

On-Demand Data Generation .....	- 2 -
Benefits at a Glance .....	- 2 -
A Danger to Client Loyalty.....	- 4 -
Too Complex to Create by Hand.....	- 4 -
Interdependencies .....	- 4 -
Customised Implementations .....	- 5 -
Historical data and logs .....	- 5 -
AI-Driven Events and Temporal Trends .....	- 5 -
Limited Data Understanding and Documentation .....	- 5 -
Key Problems to Solve.....	- 6 -
Synthetic Data Warehouses, On Demand .....	- 7 -
Intuitive, Visual Data Design .....	- 7 -
Comprehensive Functions for Complex Business Logic .....	- 8 -
Fulfilling Relationships Across the Whole Data Warehouse .....	- 9 -
Data Validation and Comparisons Build.....	- 9 -
Automatically Documenting the Data Model .....	- 10 -
Performance to Scale .....	- 11 -
Parallelisation Ensures Agility .....	- 12 -
Rapid Bug Fixes and Rigorous Testing .....	- 12 -
Transform Your Data Generation! .....	- 13 -

curiosity

# A Danger to Client Loyalty

A multi-national software vendor had launched a new, AI-driven project management tool. The flagship platform was aimed at enterprise clients, and testing and development therefore needed to mirror large enterprise implementations.

However, the vendor was struggling to test performance and functional issues, endangering relationships with their biggest clients. As clients submitted support tickets, test and development teams lacked data to replicate the bugs. This delayed defect remediation, while undermining rigorous testing of the issues in future. A lack of test data was risking customer churn.

The vendor needed a process for finding these bugs earlier, before production, while rapidly reproducing and resolving bugs in their non-production environments. However, they couldn't use live customer data and needed to generate rigorous data on demand.

## Too Complex to Create by Hand

Test environment teams were struggling to generate accurate data of sufficient variety and volumes. Whereas customers might have millions of rows in their back-end databases, testing and development was relying on databases with just hundreds.

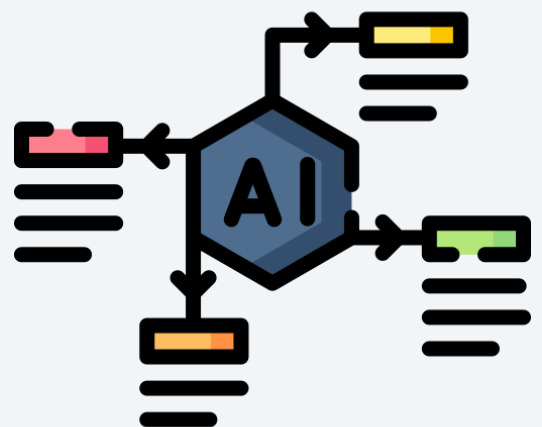
The vendor had sought to create data manually; however, this was unable to create the volume and variety of data needed to mirror enterprise software implementations. This was due to the vast and inherent complexity of the data.

## Interdependencies

The artifacts created in the project management platform must fulfil intricate interdependencies and relationships. Users cannot simply input data in isolated fields, but must navigate a hierarchy of upstream and downstream dependencies to create and edit artifacts.

The same rules apply to synthetic data generation. Data creation must produce data consistently across back-end databases. For instance, when creating data into one row of the SQL Data Warehouse, three random values might be required in three interdependent tables. Yet, these values would need to be unique across the whole system.

If a complex relationship was not fulfilled when generating data, the artifact would not appear in the front-end. It would in turn be missing from the test environment.



## Customised Implementations

Enterprise users can further customise and configure the relationships that must be fulfilled across artifacts. This adds to the complexity of generating test data, while limiting reusability.

## Historical data and logs

Non-production environments must furthermore simulate a large enterprise who has been using the platform for an extended period of time. This requires voluminous and varied data that reflects historical trends and a myriad of relationships and dependencies.

Artifacts in the project management also have associated logs. Test data cannot only reflect a snapshot of a ticket as it exists at any one moment. Instead, generated data must come with a full history of how an artifact has changed, stored in a log of past events.

System constraints further meant that the audit logs had to be managed by SQL Triggers.

## AI-Driven Events and Temporal Trends

Additional complexity arose from the sequential, temporal nature of the data, stemming from the event- and AI-based nature of the project management platform.

Test data creation must simulate chains of events. For instance, creating an artifact might spark 7 interrelated events. Data creation cannot then simply generate one artifact into an associated database in the Warehouse.

Data creation must instead generate historical data across the databases to accurately mirror a historical chain of events. Any one piece of test data must be contextualised within an array of sequential information across the system, reflecting customisable rules and AI-driven events.

## Limited Data Understanding and Documentation

While the data to generate was vastly complex, it was also poorly understood and poorly documented.

There were no specifications of the physical data model or its relationships. The only available documentation lay in Swagger specifications of the API, which is used to push data into the project management platform.

API specifications do not document the platform's back-end databases, of which the vendor had limited understanding. Test data engineers did not understand the data relationships well enough to generate data quickly and accurately. Testers and developers likewise lacked understanding of how the platform functioned in the back-end, risking unexpected behaviour and bugs.

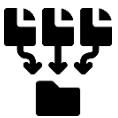
## Key Problems to Solve:



Insufficient volumes and variety of data in non-production.



Time and complexity of manual data creation.



Vast interdependencies across data and in upstream/downstream business logic.



A lack of understanding and documentation about the data.



Customisation of client implementations and the associated data.



Temporal data and series of events, both user and AI-triggered.



Numerous databases and tables needed in a functioning Data Warehouse.



Start streaming complete and compliant test data on-demand with Curiosity's Enterprise Test Data! [Talk to a Curiosity expert to get started today!](#)

# Synthetic Data Warehouses, On Demand

Today, the software vendor today leverages Curiosity's Enterprise Test Data to generate consistent data for the project management platform's artifacts, hierarchy, audit logs, and more.

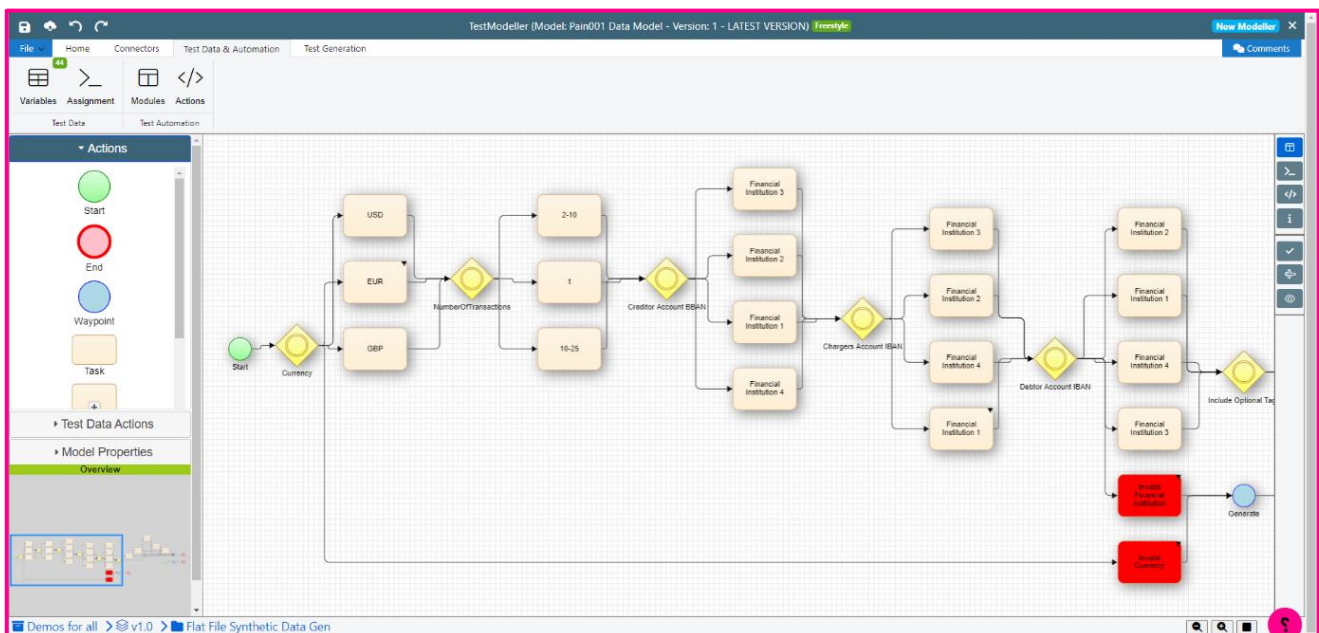
Intuitive visual flows design the data and map complex business logic, assembling comprehensive data generation functions and reusable "subprocesses". The high-performance data generation then create consistent data across tables in a SQL Data Warehouse, creating accurate test environments on demand.

## Intuitive, Visual Data Design

Intuitive, visual data design using Curiosity's workflow engine, maps the complex data and its interrelations, avoiding the time and errors associated with assembling functions in copious scripts.

The flows combine Enterprise Test Data's comprehensive, customisable data generation functions, which become fully reusable once they are configured for different data points.

- Intuitive Data Design
- Rapid Data Design
- Reusability of Work Done
- Complex Relationships and Business Logic



*Intuitive visual flows in Enterprise Test Data map data journeys through a payment application.*

The designed flows similarly become reusable, and can be applied across tables. “Nested subprocesses” link models together, embedding lower-level data generation processes within higher-level flows. The master flows map the upstream and downstream dependencies in business logic, passing parameters from one nested subflow to the next.

This intuitive, visual approach not only ensures the accuracy of data definition; it also accelerates data design and the creation of consistent data across the SQL Warehouse. Test data engineers no longer need to hand-crank a myriad of disparate processes and scripts. They can instead assemble visual building blocks to retain relationships across data.

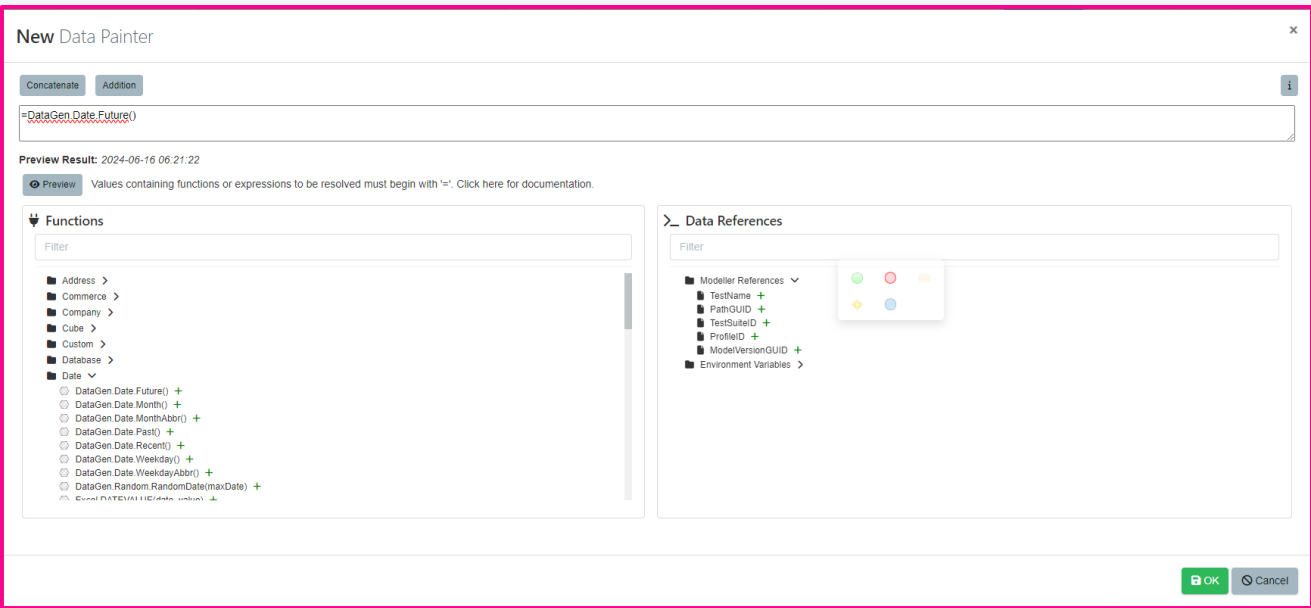
Curiosity’s automated workflow engine, then connects into the different databases, resolving the nested subprocesses in order. This passes values from one step to the next, producing consistent data that reflects complex dependencies.

## Comprehensive Functions for Complex Business Logic

The visual modelling makes it quick and simple to define complex data. For the most complex data definition, such as for historical data in artifact audit logs, the use of a visual Robotic Process Automation (RPA) engine is essential.

Enterprise Test Data’s extensive data generation functions further allow the software vendor to generate the range of data types they need. This includes the application of “event hooks”, which take data from one step in the generation process, using it in another step to reflect series of events in the project management platform.

- Functions for Every Data Type
- Reflect Event Series
- Mirror Upstream/Downstream Business Logic
- Fulfil Primary and Foreign Key Relationships



The screenshot shows the 'New Data Painter' window. At the top, there are tabs for 'Concatenate' and 'Addition'. The main input field contains the expression '=DataGen.Date.Future()'. Below this, a 'Preview Result' shows '2024-06-16 06:21:22'. A 'Preview' button is available, with a note: 'Values containing functions or expressions to be resolved must begin with "="'. The interface is divided into two main panels: 'Functions' and 'Data References'. The 'Functions' panel has a search filter and a list of functions including 'Address', 'Commerce', 'Company', 'Cube', 'Custom', 'Database', and 'Date'. The 'Date' sub-category is expanded, showing functions like 'DataGen.Date.Future()', 'DataGen.Date.Month()', 'DataGen.Date.MonthAbbr()', 'DataGen.Date.Past()', 'DataGen.Date.Recent()', 'DataGen.Date.Weekday()', 'DataGen.Date.WeekdayAbbr()', 'DataGen.Random.RandomDate(maxDate)', and 'Event.PATH()'. The 'Data References' panel also has a search filter and a list of references including 'Modeller References', 'TestName', 'PathGUID', 'TestSuiteID', 'ProfileID', 'ModelVersionGUID', and 'Environment Variables'. At the bottom right, there are 'OK' and 'Cancel' buttons.

*Enterprise Test Data provides hundreds of customisable, combinable and configurable functions, enabling flexible data generation for every data point.*



For instance, the project management platform can contain an unlimited series of dates that depend on other dates, depending on the interdependent artifacts created by a user in the tool. Enterprise Test Data's combinable date functions can easily reflect this business logic, passing a date from one function to the next while applying a wide range of logic.

## Fulfilling Relationships Across the Whole Data Warehouse

The data generation accordingly fulfils all the requisite relationships in the data. It simulates series of events in the AI-driven platform, while retaining primary and foreign key relationships throughout a generation job.

For event-based generation, Enterprise Test Data simulates events sparked by UI interactions. For instance, it generates the data created when a user creates a new ticket in the project management platform.

- Generate Data Consistently Across Every Database
- Simulate Events Sparked by UI Interactions and AI
- Functioning Warehouses On-Demand

Enterprise Test Data pushes data via the vendor's API when generating data across the back-end SQL Data Warehouse. This fulfils all the relationships needed to create a functioning Data Warehouse. The project management software is then deployed on top of this Warehouse in the non-production environment, simulating a large, historical and enterprise implementation of the software.

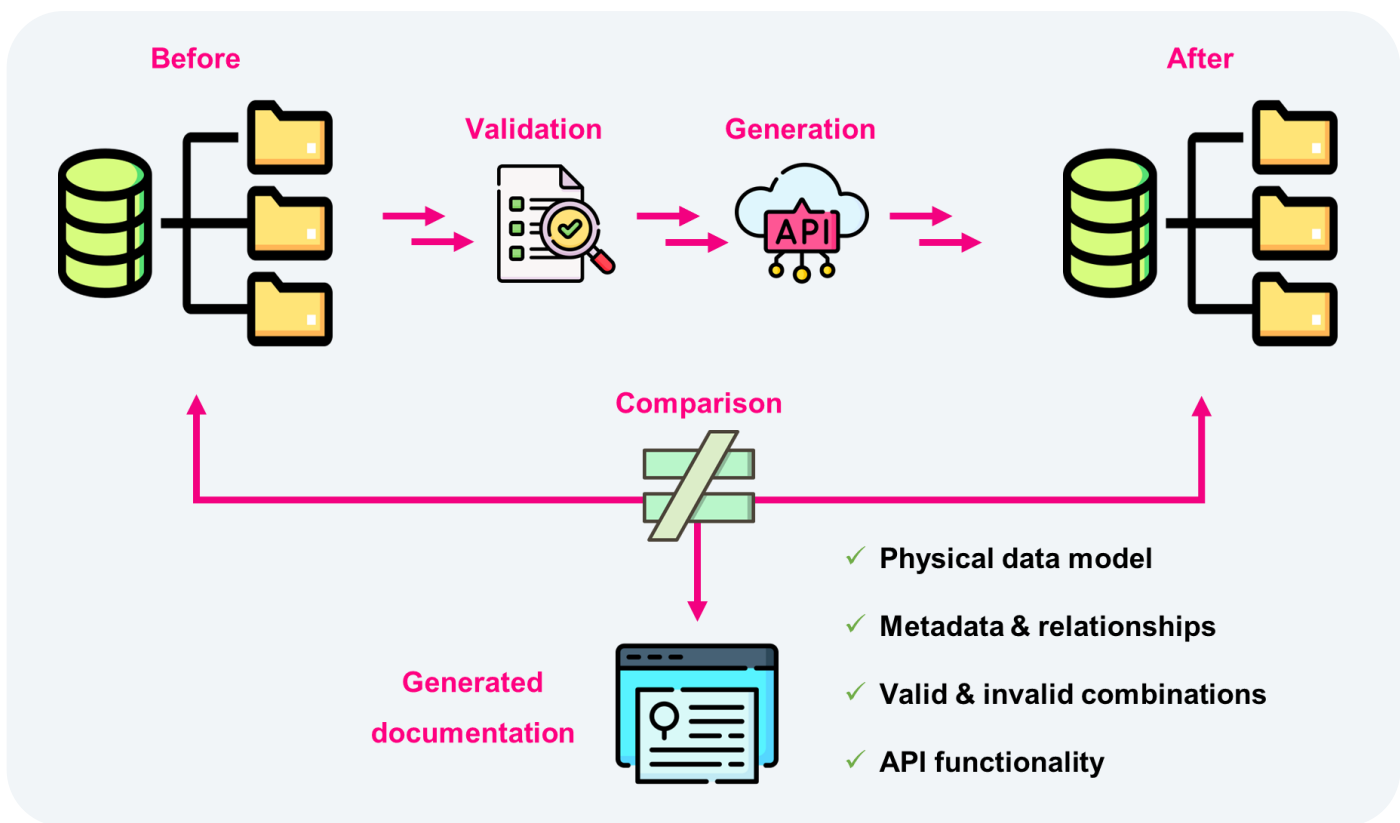
## Data Validation and Comparisons Build

While generating data via the API, automated data comparisons and validations provide understanding of the platform's complex data. They feed accurate data dictionaries and physical data models, automatically documenting the back-end databases. This enables accurate data engineering, rigorous testing, and quality development.

Data comparisons, compare the back-end databases before and after data is inputted via the API. This identifies which data has been added, updated and removed. Such modifications reveal the relationships and logic that exist within the database, revealing how data inputted via the API ripples across the back-end databases.

Automated database validations further uncover and "test" relationships in the data. They analyse the back-end data to identify which data combinations always, sometimes and never occur together. This infers primary and foreign key relationships in the data, while defining "valid" and "invalid" combinations within the data.

- Understand The Vastly Complex Data Model
- Accurate and Reusable Data Dictionaries
- Accurate API Specifications



The data validation identifies data types, groups, relationships, and more. When first implemented, it additionally discovered that the Swagger API specifications were incorrect, correcting them.

Data comparisons and validation additionally found potential bugs in the project management platform, identifying where the back-end systems were not functioning in ways expected by development.

## Automatically Documenting the Data Model

Whereas test and development teams struggled to infer database relationships and the data model from API specifications, Enterprise Test Data today automatically produces accurate documentation containing this knowledge.

The relationships and logic uncovered during data generation, comparisons and validation are automatically populated as articles in the software vendor's internal wiki.

The documentation details the metadata associated with each table, including relationships across columns and tables. The documentation additionally sets out valid and invalid data combinations, including the types of data an API call can send, and which values can be accepted by the project management platform.

- ✓ Auto-Generate and Maintain Data Documentation
- ✓ Rigorous Negative Testing
- ✓ Understand the Impact of Changes on the System
- ✓ Understand the API
- ✓ Generate Accurate Data
- ✓ Support Compliance

This documentation not only enables the generation of accurate, referentially-intact test data; data validation and comparisons also support a wide range of test and development tasks. These include:



The design and development of optimal database schemas.



Accurate platform development, understanding the API and back-end impact of systems.



Rigorous negative testing, testing with invalid combinations to protect production systems.



Data validation within DevOps pipelines to block bad data slipping into production.



The location and categorisation of sensitive data, to support compliance requirements.

## Performance to Scale

While mirroring the complex relationships and events in the historical data, Enterprise Test Data's generation provides the performance needed to create sufficient volumes of data.

Multi-threading and parallelisation boost the speed of generation,

while retaining complex relationships by generating data in many places at once. Instead of going via the front-end to simulate event-based relationships, Enterprise Test Data generates interdependent data directly to different databases in parallel.

Overall, Curiosity and Enterprise Test Data have enabled the vendor to create a gold copy data set that is 50% bigger than all their client data combined.



High-Performance Generation



Volumes and Variety to Simulate Enterprise Usage



This is a substantial advance and improvement on the hundreds of rows that the vendor could create previously by hand.

The generation routinely creates 18 million rows of data into one database, which is then propagated into 18 databases to enable parallel test and development. This includes 10 million rows of Audit Data generated for associated tickets in the project management platform.

## Parallelisation Ensures Agility

Cloning the “gold copy” data into 18 databases allows testers and developers to work side-by-side, in isolated environments. The vendor deploys the synthetic databases to cloud instances using a CI/CD pipeline, allowing testing and development without the constraints and delays caused by data cannibalisation. Different teams can work on different versions of the project management platform, in different environments, working in parallel to deliver higher quality software, faster.

# Rapid Bug Fixes and Rigorous Testing

Using Enterprise Test Data, the software vendor can now rapidly refresh test environments with complete and up-to-date test data. Whereas before testing was relying on 100s of rows of data, they now routinely generate millions of rows across multiple tables.

This data is complete in the full sense that it mirrors a production enterprise environment, including all the historical data and complex interrelationships associated with how enterprises use the project management platform. The flexible generation can furthermore be tailored to customised enterprise implementations, rapidly creating accurate environments for bug fixing and testing.



Go from slow and manual data “provisioning”, to streaming rich data in real time with Curiosity’s Enterprise Test Data! [Talk to a Curiosity expert to get started today!](#)

# Transform Your Data Generation! —



Visit [www.curiositysoftware.ie](http://www.curiositysoftware.ie) to learn more or [book a demo with a Curiosity expert today!](#)

Additionally, you can also email us at:

[info@curiosity.software](mailto:info@curiosity.software)



**Call USA:**

+1 914 218 0180



**Curiosity Software Ireland**

Unit 6 The Mill, The Maltings, Bray, Co. Wicklow, A98 XV40, Ireland

**Curiosity Software USA**

4136 Del Ray Ave. Suite 658, Marina Del Rey, CA 90292, USA

