

SOLUTION BRIEF

CA TEST DATA MANAGER AND CA SERVICE VIRTUALIZATION

CA Test Data Manager and CA Service Virtualization

Provide the on demand access to secure environments needed to deliver fully tested software on time and within budget. Generate rich virtual services with realistic, representative virtual data that covers the full range of possible scenarios, and create stable environments in which to execute every possible test. Highly distributed and outsourced teams can work in parallel, free from cross-system constraints, without exposing sensitive live data to non-production environments.

Executive Summary

Challenge

Unavailable, unfinished or constrained components can create constraints as testers and developers wait idly for components to become available “upstream.” Many organizations therefore use service virtualization in an effort to provide parallel, on demand access to the components that highly distributed teams need.

However, creating realistic virtualized services requires realistic data. Often, this is created and provisioned manually by a central team, but this is a slow and complex process and a lack of self-service access creates further delays for test teams.

Record and playback is also used as an alternative to manually creating virtual data, but this is only possible when a service already exists, creating up-stream dependencies and potential delays. Exposing live service data to non-production environments also increases the risk of a data breach and potential compliance penalties.

Where no service exists, sample data or request/response pairs must be created. This is often done by hand or by writing complex scripts. However, this is time-consuming and might not always lead to the creation of realistic functional behavior or performance. Much manual design effort is further required to engineer the correct data scenarios, as otherwise tests fail due to data inconsistency across components.

Without a more sophisticated approach, virtual service data will only represent recorded data scenarios and will not provide the outliers or future scenarios needed for rigorous testing. Defects are then detected late, and the user experience suffers. When the specification changes, virtual services can become stale and might not reflect the current message definition (API). To support new scenarios, virtual data is often maintained by hand, creating costly bottlenecks.

Opportunity

With CA Test Data Manager, virtual services that cover the full range of possible scenarios can be created, without manual data creation or maintenance. Referentially intact data can be created directly from an API specification, meaning that stable environments, free from cross-system dependencies and constraints, can be created. No live data is exposed, providing on demand environments, without risking non-compliance.

Where no service exists, virtual data can be generated synthetically from scratch, while new data can be injected into existing services to reflect the latest version of a release. Highly distributed teams are provided with on-demand access to the up-to-date environments they need to deliver fully tested software, on time and within budget.

Benefits

- Fully test software and detect defects earlier, using virtual data which covers every possible scenario.
- Avoid project delays by simulating unavailable or incomplete components.
- Create stable environments for API testing, free from system dependencies and constraints.
- Provide distributed and outsourced teams with parallel access to the secure environments they need.
- Drastically reduce pre-production infrastructure costs and avoid the risk of costly data breaches.
- Expose virtual data to distributed test and development teams in parallel and on demand.

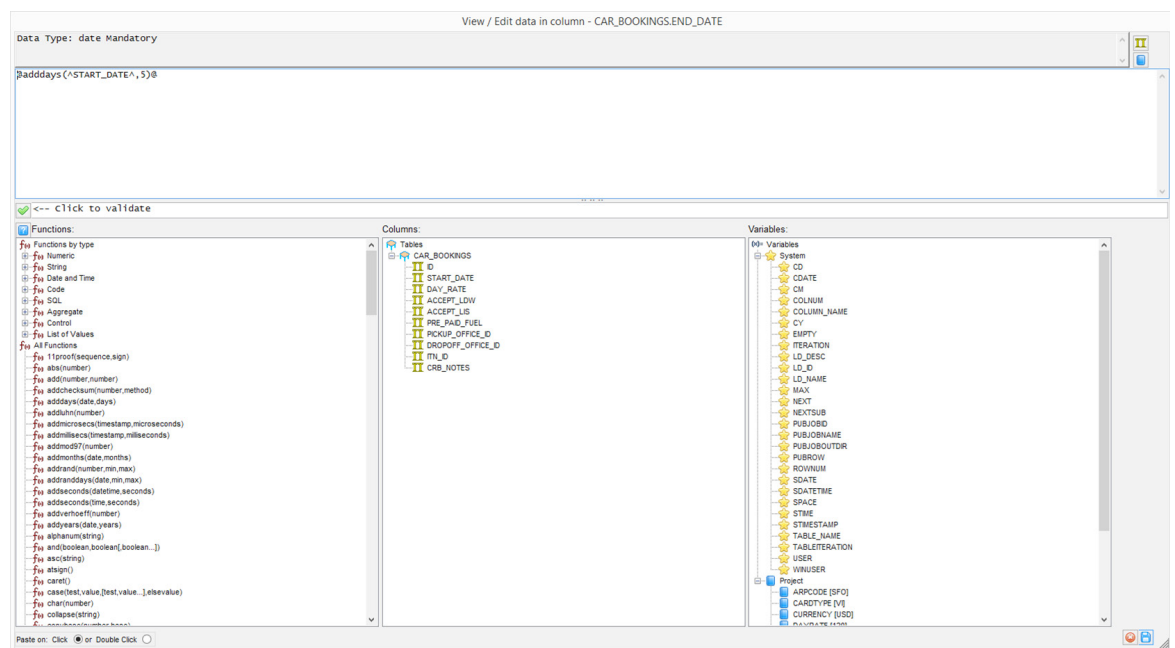
Section 1:

Synthetically Generate Request/Response Pairs Directly into a Virtual Service

CA Test Data Manager allows for the creation of the realistic virtual services needed for rigorous testing, without wasting time on manual data creation. Existing virtual data can be augmented with synthetically generated data, created directly from an API specification such as a WSDL or XML file. CA Service Virtualization can be used to “shred” an existing message definition, while sample message traffic can also be recorded and analyzed, providing the exact structure of the variables that underpin a service or message. This allows you to generate realistic synthetic request/response pairs that mirror live traffic as required for rigorous message testing of unavailable systems or components.

Figure A.

The Data Painter provides a comprehensive set of combinable functions, seed tables, system and default variables



Standard rules can be used to generate all the data associated with a given request/response pair, while a comprehensive set of seed tables, combinable functions, system and default values can be used to tailor specific data sets to test and development needs. With the in-built workflow and data orchestration engine, this is as quick and easy as selecting the target variables, shredding them and letting CA Test Data Manager automatically create a mini-database of rich virtual data in the background.

Because a parameterized approach based on variables has been taken, the virtual data can be fed directly into a deployed virtual service in CA Service Virtualization, allowing you to:

- Avoid project delays by providing testers and developers with stable environments, free from cross-system dependencies and constraints.
- Simulate unavailable or incomplete components, avoid bottlenecks created by upstream dependencies.
- Minimize the risk of costly data breaches using realistic synthetic virtual data for secure service virtualization.

Section 2:

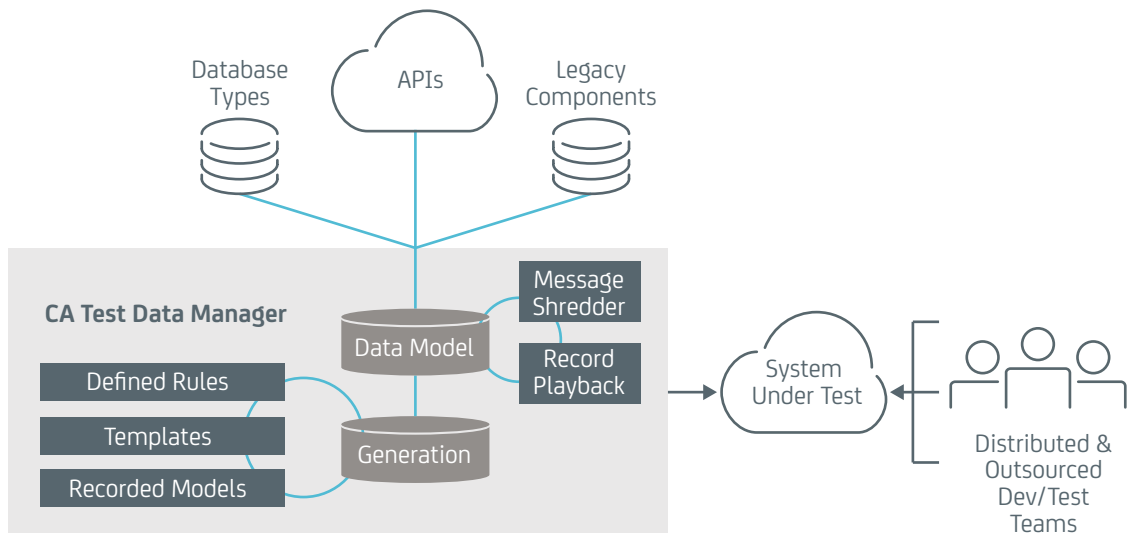
Data is Synchronized Across Components

Virtual Data is generated so that it is referentially intact across inter-dependent databases and services. Where there are available components, services and databases, CA Test Data Manager will automatically find and reserve the relevant data needed for a specific test case in minutes. At the same time, the unavailable or unfinished components can be virtualized using referentially intact, synthetic virtual data.

This means that unavailable or unfinished components can be realistically virtualized within a composite application, allowing testing and development effort to begin immediately and in parallel, without having to wait for components upstream. The meaningful data can further be fed straight into automated tests, for stable SOA testing without the delays created by automated test failure.

Figure B.

Synchronized virtual data across interdependent services, databases and components



With CA Test Data Manager and CA Service Virtualization, you can:

- Create or reserve the relevant data for a particular test across inter-dependent components
 - Feed consistent virtual data into automation engines, and avoid the delays created by automated test failure
-

Section 3:

Test the Full Range of Possible Scenarios

Existing virtual data can be augmented with synthetically generated data, while virtual data can be created from scratch so that it covers the full range of possible scenarios. This includes structured and unstructured messages, as well as dummy data for future scenarios and prototypes, providing testers with stable environments in which to execute any possible test. Unexpected results and negative scenarios can be tested against, so that defects are detected earlier, and fully tested software can be delivered on time and within budget. With CA Test Data Manager, you can:

- Generate virtual data which covers the full range of possible scenarios, as required for rigorous testing.
- Publish synthetic virtual data directly into virtualized services or automated test scripts, to execute all the tests needed to deliver quality software on time, and within budget.
- Tailor virtual data to specific test cases, using defined rules, recorded models, or templates.

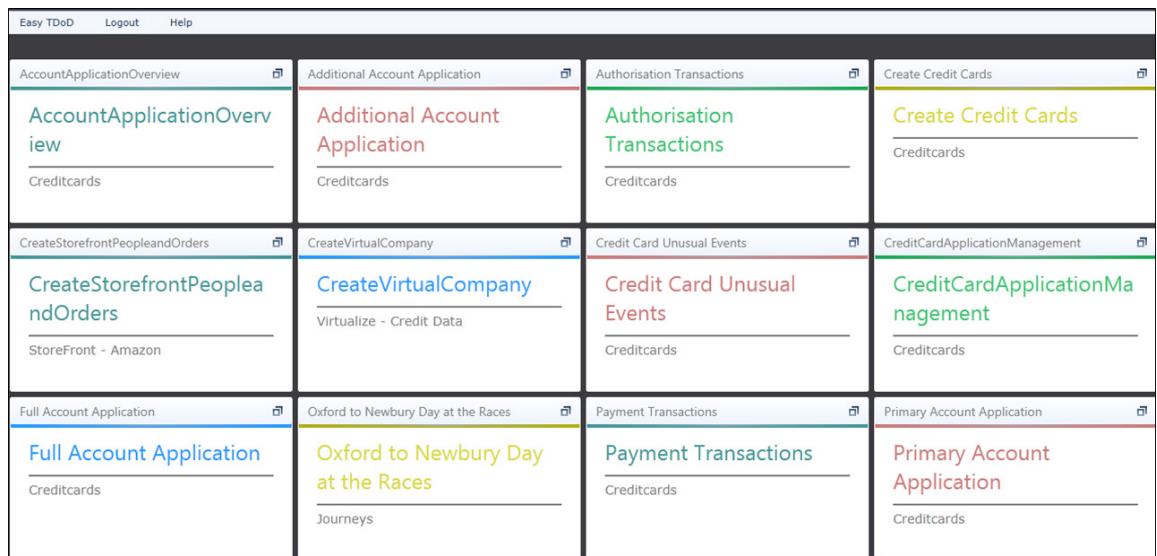
Section 4:

Expose Virtual Data to Testers on Demand

Virtual data is stored as re-usable assets in a central test data warehouse and can be exposed to test teams on demand from a self-service web portal. Request/response pairs can be cloned and received in minutes from the repository using automated data mining driven by CA Test Data Manager's in-built workflow engine. This eliminates the time wasted writing complex scripts or designing data scenarios, allowing quality software to be delivered on time and within budget.

Figure C.

An on-demand Web portal, displaying various forms that have been created to provision specific combinations of data variables



Easy TDoD Logout Help			
AccountApplicationOverview AccountApplicationOverview Creditcards	Additional Account Application Additional Account Application Creditcards	Authorisation Transactions Authorisation Transactions Creditcards	Create Credit Cards Create Credit Cards Creditcards
CreateStorefrontPeopleandOrders CreateStorefrontPeopleandOrders StoreFront - Amazon	CreateVirtualCompany CreateVirtualCompany Virtualize - Credit Data	Credit Card Unusual Events Credit Card Unusual Events Creditcards	CreditCardApplicationManagement CreditCardApplicationManagement Creditcards
Full Account Application Full Account Application Creditcards	Oxford to Newbury Day at the Races Oxford to Newbury Day at the Races Journeys	Payment Transactions Payment Transactions Creditcards	Primary Account Application Primary Account Application Creditcards

Dynamic form building means that specific combinations of data variables can be requested on the basis of specific criteria, and this is as simple as selecting variable types from drop-down menus. Testers can receive the exact request/response pair they need, “matched” to the specific tests they can execute, working in parallel to deliver fully tested systems earlier and at less cost.

Section 5

Up-to-Date Virtual Services and Data

When an API specification changes, the virtual service can be updated to support the new test scenarios required. Using CA Test Data Manager, new parameters can be injected into existing virtual services, leveraging previous effort and maximizing the value of existing virtual data. Test and development teams can be provided with up-to-date environments that reflect the latest version or release in parallel, allowing them to deliver fully tested software on time and within budget.

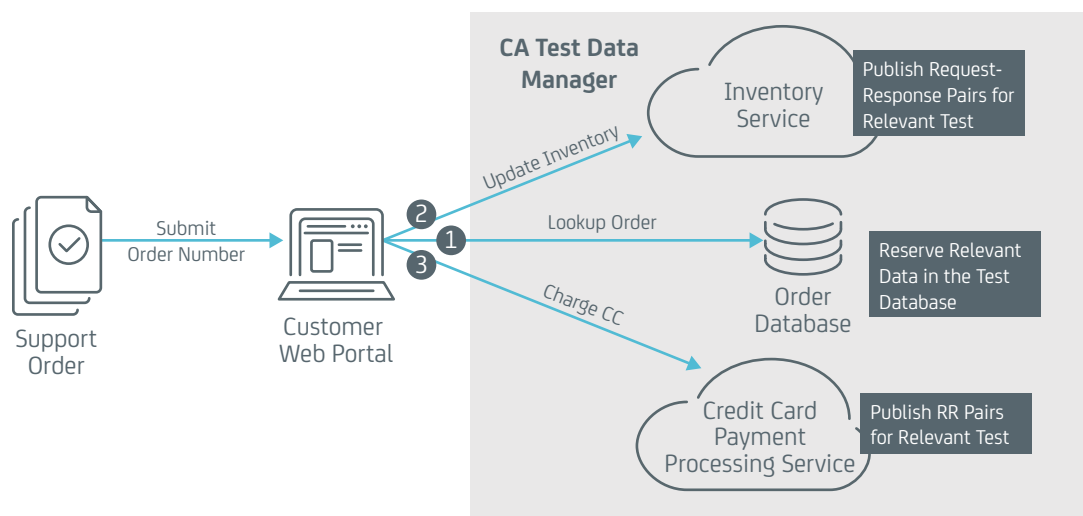
- Avoid bottlenecks created by virtual service maintenance, by injecting new parameters into existing virtual services.
- Keep up with changing user needs, by providing testers with the up-to-date environments needed to test new scenarios.
- Update virtual services to reflect new versions and releases.

Section 6

Example Use Case: Data-Driven Service Virtualization for Maximum Test Coverage

Figure D.

The scenario



In this example, the system under test is a customer web portal, used for ordering items from an online store. The web portal is part of a composite system, so that when a test order is submitted, the transaction goes through three steps:

1. An order database is searched for the order.
2. An inventory service is called to update the stock.
3. A credit card payment processing service is called to charge the credit card payment.

The Challenge

The order database is complete and available to for use by the test teams. However, the inventory service and credit card payment service are constrained. They are unavailable to the teams testing the customer web portal and need to be virtualized.

For this, virtual data needs to be created so that it is synchronized across the test databases and virtual services. This means that when a test order is submitted, the inventory service and credit card processing service must return an inventory item and a credit card that correspond to the order found in the order database. For this, synchronized request/response pairs will have to be injected into the inventory service and the credit card payment processing service. If the data is not synchronized, tests will fail due to data inconsistency, even when there is no genuine defect, creating time consuming, costly rework.

More than this, rigorous testing will require synchronized virtual data to satisfy every other possible test scenario. In a highly simplified version of this system, maximum functional coverage can be achieved with three test cases. Each test case is equivalent to submitting an order and can lead to four possible test scenarios:

1. An order is submitted, the item is in stock, and the credit card is valid. The order is therefore successful. This is a “happy path” test.
2. An order is submitted, and the item is in stock, but the credit card has expired. The order is therefore rejected, making this a negative test.
3. An order is submitted, and the credit card is valid, but the item is out of stock. The order is therefore rejected, and this is the second negative test.
4. An order is submitted, the credit card is invalid, but the item is out of stock. The order is therefore rejected, and this is the third negative test.

Data Synchronized Across Dependent Components and Services for Maximum Test Coverage

CA Test Data Manager provides the virtual data needed to cover every test. When a test is being executed, relevant data in the order database will be reserved, and request/response pairs will be generated for the dependent services. In this example, the right request/response pair will be injected into the virtual inventory service. At the same time the right data is reserved in the order database, and the right request/response pair is injected into the credit card processing system.

The synchronized reserved data and generated request/responses pairs will cover both the happy and negative scenarios outlined above, providing the environments needed to achieve 100 percent test coverage without time consuming manual processes. The constraints that arise from cross-system data dependencies are therefore lifted, so that distributed test and development teams can deliver quality software, on time and within budget.



Connect with CA Technologies at ca.com



CA Technologies (NASDAQ: CA) creates software that fuels transformation for companies and enables them to seize the opportunities of the application economy. Software is at the heart of every business, in every industry. From planning to development to management and security, CA is working with companies worldwide to change the way we live, transact and communicate – across mobile, private and public cloud, distributed and mainframe environments. Learn more at ca.com.