

CA Agile Requirements Designer – DataBuilder

Model Driven Test Data to drive your DevOps Frameworks

Create rich coherent sets of test data with cross system integrity, that lets you cover more scenarios, create stable automation layers and drive faster development.

9th May 2017

CA Agile Requirements Designer – DataBuilder

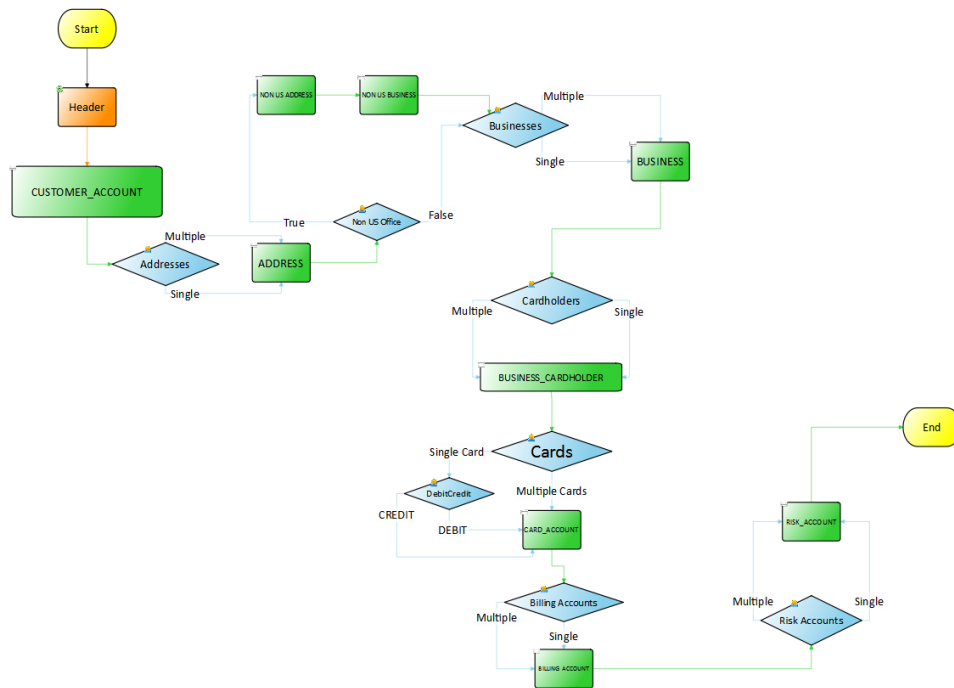
Creating coherent data at speed within increasingly faster Devops cycles is crucial to modern development practices. Whether you are modifying existing systems or creating brand new functionality having the correct test data to drive high quality development and testing is crucial to the success of all projects.

CA Agile Requirements Designer – DataBuilder

Introducing DataBuilder - a toolset specifically designed to find and create the correct data for cross system testing. DataBuilder is built on the core technologies of CA Test Data Manager and CA Agile Requirements Designer. By combining the power of advanced modelling with the world's most powerful test data generation engine DataBuilder can very quickly improve the quality and speed of your development.

DataBuilder is fully integrated with the worlds' leading software lifecycle tools such CA Agile Central, HPE ALM and Jira allowing you to attach and define the data you need to the user stories as well as auto generating matching test cases.

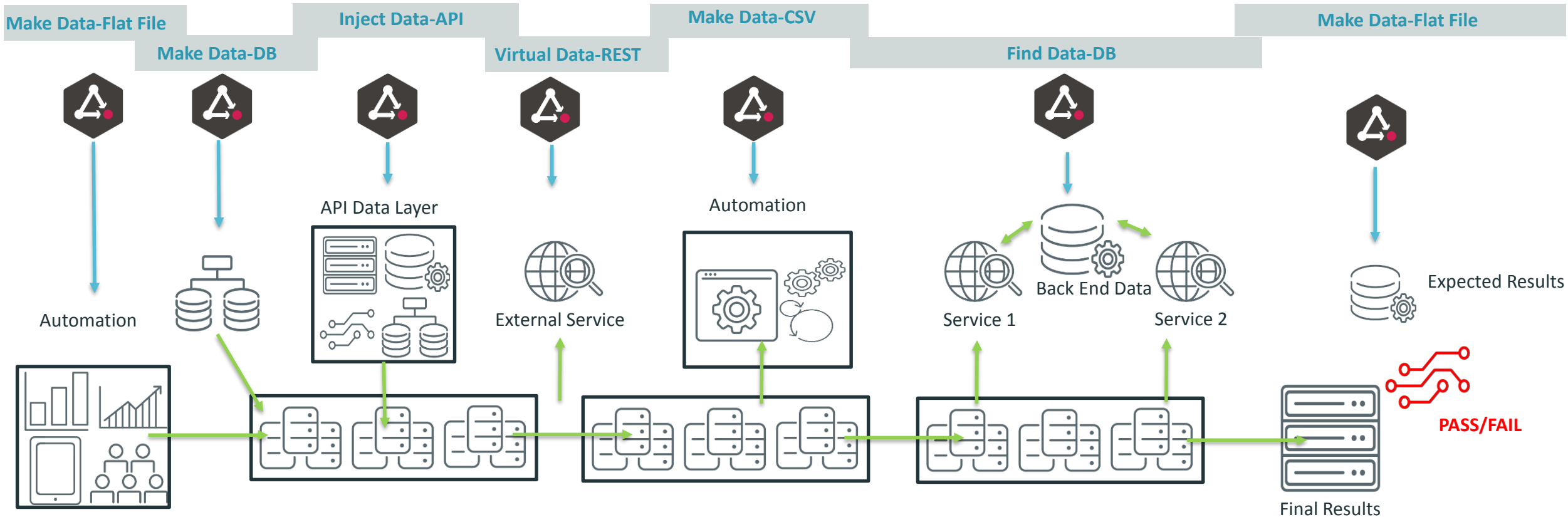
DataBuilder – Using Modelling to Cover Combinations of Data



test_name	db_ADDRESS_REPEATER	db BILLING_ACCOUNT_REPEATER	db BUSINESS_CARDHOLDER_REPEATER	db BUSINESS_REPEATER	db_CARD_ACCOUNT_REPEATER	db_CUSTOMER_ACCOUNT_REPEATER	db_RISK_ACCOUNT_REPEATER	tdCARDNAM
170 AllComb170	1	2	2	1	1	1	1	CREDIT
171 AllComb171	1	1	2	1	1	2	2	CREDIT
172 AllComb172	1	1	2	1	1	1	1	CREDIT
173 AllComb173	1	2	2	1	1	2	2	DEBIT
174 AllComb174	1	2	2	1	1	1	1	DEBIT
175 AllComb175	1	1	2	1	1	1	2	DEBIT
176 AllComb176	1	1	2	1	1	1	1	DEBIT
177 AllComb177	1	2	2	1	4	1	2	--EMPTY--
178 AllComb178	1	2	2	1	4	1	1	--EMPTY--
179 AllComb179	1	1	2	1	4	1	2	--EMPTY--
180 AllComb180	1	1	2	1	4	1	1	--EMPTY--
181 AllComb181	1	2	1	1	1	1	2	CREDIT
182 AllComb182	1	2	1	1	1	1	1	CREDIT
183 AllComb183	1	1	1	1	1	1	2	CREDIT
184 AllComb184	1	1	1	1	1	1	1	CREDIT
185 AllComb185	1	2	1	1	1	1	2	DEBIT
186 AllComb186	1	2	1	1	1	1	1	DEBIT
187 AllComb187	1	1	1	1	1	1	2	DEBIT
188 AllComb188	1	1	1	1	1	1	1	DEBIT
189 AllComb189	1	2	1	1	4	1	2	--EMPTY--
190 AllComb190	1	2	1	1	4	1	1	--EMPTY--
191 AllComb191	1	1	1	1	4	1	2	--EMPTY--
192 AllComb192	1	1	1	1	4	1	1	--EMPTY--

Once you have built an initial model of your required data scenarios you can quickly advance to auto-generating much more sophisticated sets of combinations of data values. So from a single model you can guarantee you are hitting all the potential combinations of data.

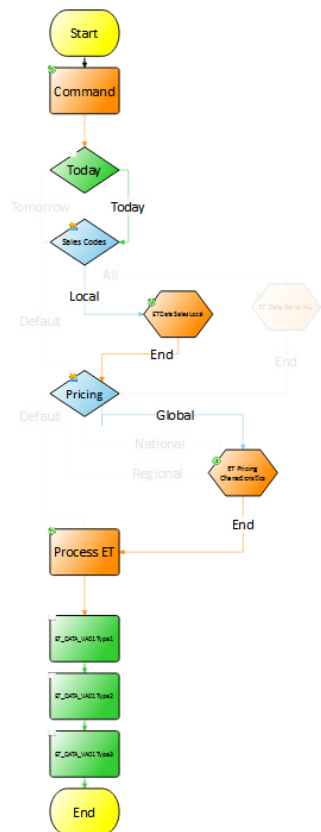
Creating Test Data to support complex chains of data interactions across multiple systems



DataBuilder can find or create test data that is synchronized and coherent across multiple systems. Most testing nowadays requires that data in other dependent systems is linked to your test case. Setting up cross system data is very time consuming and fragile, in some cases it can appear impossible.

DataBuilder lets you model the cross system data dependencies and then either find and reserve test data or create new, matching data. The data can be injected into CA Service Virtualization to create “intelligent” responses, message data, directly into databases or via automation scripts.

By using simple to understand models users can drag and drop data characteristics and quickly create complex test data scenarios



Block	Test Data
Start	
Command	
Today = Today	db_CREATION_DATE = ~SDATE~
Sales Codes = Local	
ET Data Sales Local = End	db_SALES_DISTRICT = ^ET Data Sales Local::db_SALES_DISTRICT^; db_SALES_GROUP = ^ET Data Sales Local::db_SALES_GROUP^; db_SALES_OFFICE = ^ET Data Sales Local::db_SALES_OFFICE^
All	
Sales District = Local	db_SALES_DISTRICT = Local
Sales Group = HQ	db_SALES_GROUP = HQ
Sales Office = NY	db_SALES_OFFICE = NY
End	
Pricing = Global	
ET Pricing Characteristics = End	db_PRICE_GROUP = ^ET Pricing Characteristics::db_PRICE_GROUP^; db_PRICE_LIST_TYPE = ^ET Pricing Characteristics::db_PRICE_LIST_TYPE^; db_PRICING_REFERENCE_MATERIAL = ^ET Pricing Characteristics::db_PRICING_REFERENCE_MATERIAL^
Global	
Price Group = Global	db_PRICE_GROUP = Global
Price List Type = Combo	db_PRICE_LIST_TYPE = Combo
Pricing Reference = Direct	db_PRICING_REFERENCE_MATERIAL = Direct
End	
Process ET	
ET_DATA_VA01 Type 1	db_ET_DATA_VA01_REPEATER = 1; db_SALES_DOCUMENT_ITEM = 1
ET_DATA_VA01 Type 2	wk_TYPE2_Repeater = 3
ET_DATA_VA01 Type 3	db_ET_DATA_VA01_REPEATER = 1; db_SALES_DOCUMENT_ITEM = 3
End	

Data for negative testing can easily be created by utilizing sub flows of common invalid data scenarios.

If data for tests cannot be found DataBuilder will create the missing data.

As data is created or found it can be used to drive automation frameworks

Automation relies heavily on stable test data, much time can be spent manually creating test data or looking for data related automation issues. With DataBuilder you can track the keys to drive the automation as well as storing the expected results.

The effect is to allow automation frameworks to be simpler with test data being created before they run and expected results fed into the framework.

Who would use DataBuilder and for What Project Types?

DataBuilder can create data for users for acceptance testing, for testers to manually test with and for developers to develop against. Good rich test data is required right across all aspects of a dev cycle, including: design; development; user, integration, regression and performance testing as well as for user training.

Typical projects that require complex data to be created and maintained are:

- Big Data testing
- ETL testing
- Migration testing
- Business Intelligence testing
- EDI File processing
- Time shifted data

CA Agile Requirements Designer – DataBuilder

Simple to use Test Data Generation and Data Modelling.

Driving Continuous Development and DevOps frameworks.