

Using TDM 4.0 with DevTest 10.0

Daniel Mazzei, Associate Engineering Services Architect

January 2017

© 2016 CA. All rights reserved. CA confidential & proprietary information. For CA, CA Partner and CA Customer use only. No unauthorized use, copying or distribution. All names of individuals or of companies referenced herein are fictitious names used for instructional purposes only. Any similarity to any real persons or businesses is purely coincidental. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies. These Materials are for your informational purposes only, and do not form any type of warranty. The use of any software or product referenced in the Materials is governed by the end user's applicable license agreement. CA is the manufacturer of these Materials. Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-19(c)(1)-(2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Introduction

- What?
 - TDM integration with DevTest lets you update running virtual services with realistic, representative data to cover a wide variety of the possible testing scenarios
- How?
 - We push the generated Request/Response (RR) pairs into a running virtual service, which augments the virtual service with the synthetic data similar to production data
 - Then we route application traffic to the VS to test the application

Introduction

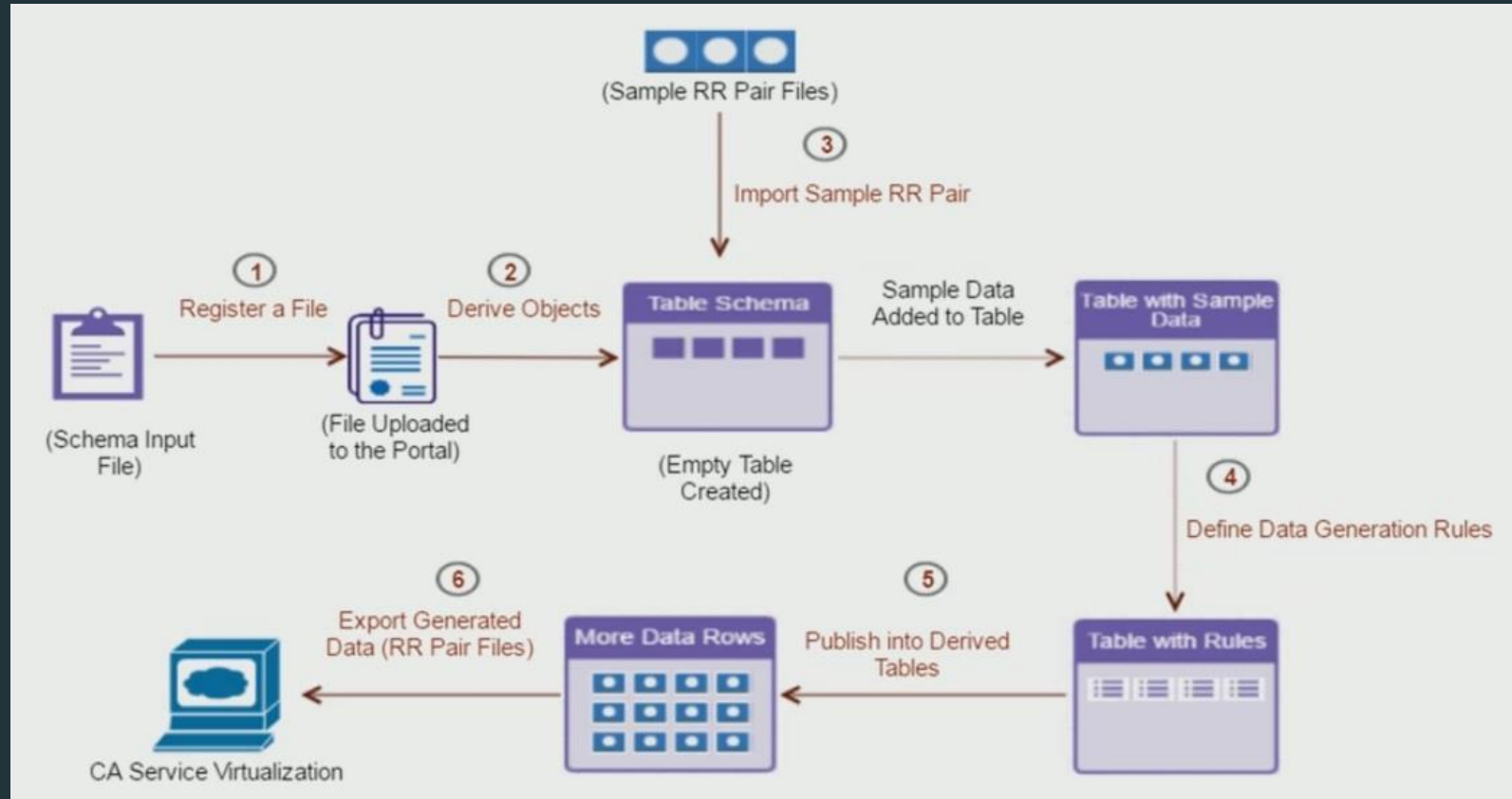
■ Why?

- Maintains referential integrity of data by creating it directly from an API specification (e.g., WSDL), which helps create stable environments that are free from cross-system dependencies
- Removes the need to create or maintain data manually for virtual services
- Avoids project delays by simulating unavailable or incomplete components
- Provides uninterrupted access to current environments to distributed teams, which enables them to continually deliver fully tested applications
- Provides on-demand environments without risking non-compliance because no live data is exposed

Prerequisites/Assumptions

1. Create and select a project and version in TDM Portal
2. Create data generator in TDM Portal
3. Create connection profile to DB in TDM Portal
4. Configure TDM Portal with DevTest Portal connection info
5. Create and start a virtual service in DevTest Portal
 - Using “Virtualize using RR pairs” method, which requires request and response XML files

TDM Data Creation Process



Data Creation Process

1. Register file object in TDM so we can perform data generation
 - Modeling → Objects → Register New Object
 - Must be WSDL, XML RR pair, JSON RR pair, or REST RR pair
2. Create and register derived objects in the target database based on the registered file object
 - Click on created object → Select Connection Profile → Create and Register Tables
 - To verify tables were created: Modeling → Objects → Click on object
 - To verify tables are empty: Generators → Click on generator name → Click on Registered Tables → Select all to show empty

Data Creation Process

3. Import data (sample R/R pairs) into derived object

- Modeling → Objects → Click on object → Click Import Data icon → Enter RR Pair Link ID → Select RR Pair Files → Select Import to Generator → Click Import
- To confirm import successful: Generator → Click on generator name → Tables should include imported data

Data Creation Process

4. Define data generation rules to create synthetic data
 - Select All Tables → Click Relational Edit button → Select both checkboxes so that we
 - Automatically convert values to expression with unique sequential number when publishing
 - Establish foreign references in related tables where primary key is referenced
 - Select all tables from list
 - Click on additional fields and use Data Painter to write rules
 - Use RANDLOV function with ZIP seed list to generate random zip codes
 - Edit expression and Validate
 - Write rules for remaining columns (e.g., address, telephone, etc.)

Data Creation Process

5. Publish data into derived objects

- Click Publish → Select all tables → Select DB Schema and Connection Profile → Set Repeat to 10 → Click Publish

6. Export generated data (as RR pairs) and publish to DevTest

- Modeling → Objects → Click on object → click on Export Data icon → Select Update Virtual Service checkbox → select VSE and VS → Export
- Verify VS was redeployed and transaction count reset to zero
- Verify correct RR pairs uploaded to VS by sending Request and comparing Response

Best Practices

■ Creating Virtual Services

- In DevTest, you can create a virtual service using multiple methods (e.g., via recording and RR pair). However, *TDM is not dependent on the method that you used to create the virtual service. TDM can update it with generated RR pairs regardless of the method that you used to create the service.*

■ Performing Multiple Operations

- (For WSDL) You can use one file object to create and register derived objects only for one operation at a time. *To use the same file object for multiple operations, register that file object separately for each operation and create and register derived objects.*

Best Practices

■ Identifying Derived Objects

- If you use the same database as a target connection profile to derive objects for the different file objects, it becomes difficult to identify which derived objects belong to which file object. In this case, you can *uniquely identify derived objects based on the schema name*.

■ Adding Unique Values to Primary Keys

- Every derived object includes a column that is named Shred_ID, which must include a unique value. TDM recognizes this ID as an identity of a derived object. Instead of manually adding a unique value to this primary key column, *use the Make All Parents Default option in the CA TDM Portal to automatically add this value during data generation*.

Best Practices

- Using Correct RR Pair Naming Convention
 - Ensure that the RR pair files that you use for uploading the data into a virtual service *follow the correct naming convention* (e.g., TDMFile_1655-Req.xml and TDMFile_1655-Rsp.xml).
- Using the Same Connection Profile and Schema Name
 - *Do not change the connection profile and schema names* that you used at the time of creating and registering derived objects.



Daniel Mazzei

Engineering Services Architect

Daniel.Mazzei@ca.com

