TDM (TEST DATA MANAGEMENT) UPGRADE PROCEDURE TO V8.0

This document describes the following:

How to upgrade TDM onto the present version: **V8.0.**

How to re-implement the modified product features.

Notes:

This document does not cover the Fabric server topology changes, such as additions of nodes, data centers, changes of replication factors or consistency level.

The TDM upgrade procedure should be performed on testing environments prior to applying it on your production deployment.

Perform a sanity test upon completion of the upgrade procedure, such as running a few TDM tasks and conducting other checks per the sanity procedure defined in your project.

SOFTWARE UPGRADE PROCEDURE

1. TDM Installation - Prerequisites

The following components should be installed as a prerequisite:

Fabric Server - Fabric 7.1 and above for TDM 8.0.

PostgreSQL DB - the TDM DB tables are created on a PostgreSQL DB. TDM V8.0 supports v9.6 and above (TDM V8.0 was certified based on v13).

2. Related Documents

FABRIC UPGRADE PROCEDURE TO V7.1.0

3. Installation of Fabric Server

Download Fabric docker from the download page.

Note:

Fabric server no longer has the TDM directory under ~/ webserver/static directory. The TDM portal code is included in the TDM library and is deployed from the TDM project.

For more information about TDM V8.0 installation, please read the TDM Installation article in the TDM Configuration.

4. Add the Synthetic Environment

Open the TDM Fabric project in Fabric Studio and add the **Synthetic** environment to the project's Environments. The Synthetic environment must be created and deployed in Fabric with the TDM and DB_CASSANDRA interfaces (these interfaces must be enabled). Note that the Synthetic environment is created in the TDM DB by the TDM DB upgrade script.

5. Import the Updated TDM V8.0 Library

Step 1 – Back Up and Update the TDM Project Before the Import

- Back up and export the Fabric project. Verify that you have backups for the following objects:

  CustomLogicFlow.actor

  TDMSeqList.actor

  TDMSeqSrc2TrgMapping.actor

  FilterOutTable.actor

  TDMTargetTablesNames.actor
- Delete the following:
  - TDM Library's Java files under the Shared Objects.
  - TDM Library's Shared Broadway flows and Templates.
  - TDM Library's Web Services.
  - TDM LU.
  - TDM_LIBRARY LU.

Step 2 - Import the TDM Library to the Project

Open the TDM Fabric project in Fabric Studio and custom import the following objects of the updated TDM 8.0 Library into the Fabric project:

- Web Services: import all the Web Services' files.

- Shared Java files: import all Java files in the Shared Objects.

- Shared Broadway Flows and Actors:
  - Import the shared Broadway flows and Actors from the TDM Library.
  - After the import, edit the following Actors:
    - **TDMSeqList**:
      - Custom import the backed-up object.
      - If you use the FabricRedis interface to get the next value from Redis, it is recommended to do the following in order to use a newly created DB sequence (created in the TDM DB) instead of Redis:
        - Update the SEQUENCE_REDIS_OR_DB setting to TDM.
        - Update the Init inner flow for the sequence: use the COALCASE function when getting the initial value for the newly created sequence. For example:

```
Select COALESCE(max(activity_id),0) + 100000 as init_activity_id from activity;
```

        - Regenerate the sequence Actors to create and use the sequence in the TDM DB.

      Click here for more information about the sequence implementation.

    - **TDMSeqSrc2TrgMapping**:
      - Custom import the backed-up object.

    - **TDMFilterOutTargetTables:**
      - Custom import the backed-up object.
      - Open TDMFilterOutTargetTables.actor for editing. A new column was added: generator_filterout with type Boolean. Check it for all the TDM product tables and the _TAR table, setting it to **true**.

    - **CustomLogicFlows:**
      - Custom import the backed-up object.

- Shared Templates: import the shared templates from the TDM Library.

- Shared Translations:
  - Edit the **trnRefList**: update the id's type to Auto Increment.

- Shared Java files: import the shared Java files from the TDM Library.

- Shared Globals:
  - Import the shared Globals file from the TDM Library.
  - Import and override the shared Globals from the backed-up project to merge the project's setting and Globals into the shared Globals.
  - A new shared Global has been added in TDM 8.0 to limit the number of entities in the TDM execution reports: **TDM_SUMMARY_REPORT_LIMIT**. The default value is 10,000. Edit it if needed.

- TDM_LIBRARY LU:
  - Import the TDM_LIBRARY LU from the TDM Library.

- TDM LU:
  - Import the TDM LU. Use the TDM LU export file (it contains only the TDM LU).
  - Note that you must use the *Import All* option to import the TDM portal code under the web subdirectory. The Custom import method does not import the TDM portal code as the web subfolder is not supported by the desktop studio (it is supported only by the web studio).

- TDM_Reference LU:
  - Import the TDM_Reference LU from the TDM Library.

Step 3 – Update the LUs

- Copy and override the LU tables from the TDM_LIBRARY LU into each LU. This is needed since TDM 8.0 moved the TDM enrichment functions to LU table's Broadway population flows.

○

- Replace DB Queries and Root function LU populations with Broadway flow populations to enable a data generation implementation.
- Regenerate the population on the main source LU table based on the populationRootTable template.

Step 4 – Deploy:

- Redeploy the Web Services to Fabric.
- Redeploy the TDM LU to Fabric:
  Verify that the **BUILD_TDMDB** shared Global is set to **false** before deploying the TDM LU, otherwise the TDMDB will be created in the deploy flow.
- Redeploy the LUs to Fabric.
- Redeploy the TDM_Reference to Fabric.

6. TDMDB Upgrade

Soft deploy the TDM LU to Fabric debug server. Before the deploy, verify that the TDM interface is updated with the TDM DB connection details.

Run the **RunTDMDBUpgradeScripts** flow. Populate the current version and the target version input parameters. Set the target version parameter to 8.0. For example:

CURRENT_TDM_VERSION = 7.6.

TARGET_TDM_VERSION = 8.0.

Note that the version's list is set in TDMDBUpgradeScripts Actor (imported from the TDM Library).

7. Data Generation Implementation

Add the data generation implementation.

Click here for data generation's implementation guidelines.