TDM UPGRADE PROCEDURE

TDM (TEST DATA MANAGEMENT) UPGRADE PROCEDURE TO V7.6

This document describes the following:

- How to upgrade TDM onto the present version: **V7.6.**
- 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 7.0 and above for TDM 7.6.
- **PostgreSQL DB** - the TDM DB tables are created on a PostgreSQL DB. TDM V7.6 supports v9.6 and above (TDM V7.6 was certified based on v13).

2. Related Documents

- FABRIC UPGRADE PROCEDURE TO V7.0

3. Installation of Fabric Server

- Download Fabric docker from the download page.
- Notes:
- All Fabric nodes should work with the same Redis instance. This requirement will result in an alignment of the sequence and in masking values generated by the TDM tasks. Open the Redis port on one Fabric node and run Redis.
- The TDM and Redis interfaces are defined in the Fabric project.
- Fabric server no longer has the TDM directory under ~/ webserver/static directory. The TDM GUI is included in the TDM library and is deployed from the TDM project.

For more information about TDM V7.6 installation see the TDM Installation article in the TDM Configuration.

4. Import the Updated TDM V7.6 Library

Backup the Fabric project. Then take the following steps:

Step 1 - Update the TDM Project Before the Import

- Web Services:
  Delete all the TDM product files from the Web Services.

- Translations:
  - Save a backup of the trnRefList object.

- Shared Logic:
  - Delete the product logic files.

Step2 - 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 7.6 library into the Fabric project:

- Web Services:
  Import all the Web Services' files from the TDM library.

- Shared Interfaces:
  - Import the POSTGRESQL_ADMIN interface and verify that it is inactive.

- Shared Broadway Flows and Actors:
  - Import the shared Broadway flows and actors from the TDM library. Make sure the following actors are not overridden:
    - TDMSeqList
    - TDMSeqSrc2TrgMapping
    - TDMFilterOutTargetTables
    - CustomLogicFlows

- Shared Templates:
  - Import and override the shared templates from the TDM Library.

- Shared Translations:
  - Import trnRefList translation object from the TDM library.
  - Populate the trnRefList based on the backup file, saved in step1.

- Shared Functions:
  - Import the shared Logic files from the TDM Library.

- Shared Globals:
  - Import and override the shared Globals file from the TDM Library.

- TDM and TDM_LIBRARY LUs:
  - Import and override the TDM and TDM_LIBRARY LUs.
  - Edit the deploy.flow of the TDM LU:
    - Edit the Set Env (FabricSet) actor of the Set Environment stage: populate the value of the environment key with the environment name from which the deploy.flow needs to get the Cassandra and Redis connection details.

- TDM_Reference LU:
  - This is a new LU, added in TDM 7.6 for reference handling. Import it from the TDM Library.

Step 3 – Update the Custom Logic Flows

- The input parameter name of **CheckAndStopLoop** actor has been updated from recordLoaded to RECORD_LOADED. Open the custom logic flows in the project. On each flow:
  - Re-link the CheckReserveAndLoadToEntityList flow's output to RECORD_LOADED input parameter of the **CheckAndStopLoop** actor**.**
  - Verify that the **NUM_OF_ENTITIES** input parameter is set as **an external input** in the **CheckAndStopLoop** actor**.**

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 in order to prevent the TDMDB recreation by the TDM LU's deploy flow.
- Redeploy the LUs to Fabric.
- Deploy the TDM_Reference to Fabric.

5. 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 7.6. For example:
- CURRENT_TDM_VERSION = 7.4.
- TARGET_TDM_VERSION = 7.6.
- Note that the version's list is set in TDMDBUpgradeScripts actor (imported from the TDM Library).

6. Reference Upgrade

- Previous TDM versions stored the extracted Reference tables in the Cassandra DB. However, TDM 7.6 stores the Reference tables in Fabric, in a dedicated LU: TDM_Reference (each Reference table is stored as a separate LUI). When running Extract or Load tasks on Reference tables, the first execution gets the Reference data and syncs it into Fabric.
- Notes:
- The execution of the following tasks, created on previous TDM versions, fails in TDM 7.6:
- Load task for *Reference and Entities* with the sync policy set to 'Do Not Sync from Source Data'. The execution of this task fails as it cannot sync the Reference tables into Fabric due to the task's sync policy. Therefore, it is recommended to edit these tasks and set the sync policy to 'Sync New Data' before the first execution.
- Data versions on Reference tables, created by previous TDM versions, are not in use in TDM 7.6.
- The first task executions with Reference table may take time since they need to sync the Reference tables into Fabric.