TDM (TEST DATA MANAGEMENT) UPGRADE PROCEDURE TO V8.1

This document describes the following:

- How to upgrade TDM onto the present version: **V8.1.**
- How to re-implement the modified product's 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 7.2 and above for TDM 8.1.
- PostgreSQL DB - the TDM DB tables are created on a PostgreSQL DB. TDM V8.1 supports v9.6 and above (TDM V8.1 was certified based on v15).

2. Related Documents

FABRIC UPGRADE PROCEDURE TO V7.2.0

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 V8.1 installation, please read the TDM Installation article in the TDM Configuration.

4. Import the Updated TDM V8.1 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 8.1 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 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.
  - New shared Globals have been added in TDM 8.1. Edit them if needed:

Global name	Description	Default value	Additional information
TDM_DELETE_TABLES_PREFIX	TDM 8.1 added an automatic generation of the target LU table to support the deletion of an entity by a TDM task. This Global defines the prefix for the generated target LU tables.	_TAR	Delete entities implementation
TDM_BATCH_LIMIT	Limit the number of entities to be populated into the TDM execution tables for a task execution. By default, the number of entities for a	-1	Task execution process

Global name	Description	Default value	Additional information
	task execution is unlimited (-1)		
TDM_SEQ_REPORT	When set to true (the default value), the task execution populates the TDM_SEQ_MAPPING table and generates the Replace Sequence Summary Report tab in the task execution report. This Global can be set to false to avoid the population of TDM_SEQ_MAPPING and the generation of the Replace Sequence Summary Report for a better performance.	true	TDM Flows Implementation
TDM_POPULATE_JMX_STATS	Set this Global to **true** to enable the execution of **populationJMX** population flow on TASK_EXECUTION TDM LU table.	false	TDM statistics handling
SEQ_CACHE_INTREFACE	Starting from Fabric V7.2, SQLite and PostgreSQL are also supported as Fabric operational DBs in addition to Cassandra. If you use the TDM PG DB as the operational (system) DB, update this Global accordingly and set it to TDM.	DB_CASSANDRA	Fabric operational database

- **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.
      - Note that starting from TDM 8.1 the CACHE_DB_NAME field is removed from the tdmSeqList, since the caching DB is taken from the SEQ_CACHE_INTREFACE Global. The old CACHE_DB_NAME will no longer be checked when generating sequence Actors. Instead, verify that the SEQ_CACHE_INTREFACE Global is set properly.

Click here for more information about the sequence implementation.

Click here for more information about Fabric operational DB.

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

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

- **CustomLogicFlows:**
  - Custom import the backed-up object.
  - Open the object for editing. A new Boolean field has been added: DIRECT_FLOW. Add this field and leave it cleared for the existing records.
  - Add a new record to the CustomLogicFlows table:
    - LU_NAME: will be empty
    - FLOW_NAME: CustomLogicSql
    - DESCRIPTION: Generic Custom Logic Flow for Sqls
    - DIRECT_FLOW: true

See example:

CustomLogicFlows : table

	LU_NAME (string)	FLOW_NAME (string)	DESCRIPTION (string)	DIRECT_FLOW (boolean)	+
1	Customer	get5GCustomers	Get 5G customers by the contract status	☐	✗
2	Customer	getCustomersByAddress	Get customers by their state	☐	✗
3	Customer	getCustomersBySQL	Get customers by input SQL and input parameters on the CRM	☐	✗
4	Customer	getCustomersBySQL2	Get customers by input SQL on the CRM DB	☐	✗
5	Customer	getAll5GCustomers	Get all 5G customers	☐	✗
6		CustomLogicSql	Generic Custom Logic Flow for Sqls	☑	✗

Click here for more information about Custom Logic implementation.

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

- **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 FABRIC_TDM_ROOT LU and LU_PARAMS tables from the TDM_LIBRARY LU into each LU. This is needed due to the following TDM 8.1 changes:

- FABRIC_TDM_ROOT table – the population flow has been updated to get the root entity id for each LUI, set it into a session key level, and populate it into task_execution_entities and <LU Name>_params TDM DB tables.
- LU_PARAMS – both, table's structure and table's population, have been changed. Note that TDM 8.1 does not require adding the LU parameters to the LU_PARAMS table.

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.

5. Run the Upgrade Flow

The upgrade flow needs to run in order to upgrade the following:

- TDM database
- Convert the TDM translations into MTables

BACK UP THE TDM DB

Back up the TDM DB. You can use the PGAdmin and create a Backup file on the TDMDB.

TDM DB UPGRADE

- The **RunTDMDBUpgradeScripts** flow runs update_tdmdb_from_v8.0_to_v8.1.sql script. This script adds new fields to the TDM DB tables and populates the new fields in task_execution_entities and <LU Name>_params tables.

- Note that previous TDM versions created a materialized view in the TDM DB on each combination of BE and source environment to enable a selection of entities based on parameters. From TDM 8.1 onwards, each LU parameters table (naming convention: <LU name>_params) contains the following fields to connect the entity id to its root entity:
  - root_lu_name
  - root_iid
- The materialized views are no longer needed in TDM 8.1. Still, to be on the safe side, **the upgrade script does not drop the materialized views from the TDM DB**. You can drop them after running the TDM 8.1 successfully for a while.

CONVERT THE TDM TRANSLATIONS INTO MTABLES

- TDM 8.1 replaces the previous TDM translation with MTables to support a development of the TDM on both Fabric Studios - Desktop-Studio and Web-Studio.
- The **RunTDMDBUpgradeScripts** flow runs the **convertLuTranslations** flow to convert old TDM translations to the equivalent TDM MTables. Each execution of the convertLuTranslations flow deletes and re-populates the related MTables. **Note that**

all LUs must be deployed to Fabric debug server before and after running the upgrade flow.

TDM UPGRADE FLOW EXECUTION

- Deploy all LUs to Fabric debug server.
- Deploy the TDM LU to Fabric debug server. Before the deploy, verify that the TDM interface is updated with the TDM DB connection details.
- Verify that the imported **TDMDBUpgradeScripts** Actor has an entry to upgrade the TDM to 8.1: verify that the script_name is populated with update_tdmdb_from_v8.0_to_v8.1.sql and the flow_name is populated with convertLuTranslations.
- Run the **RunTDMDBUpgradeScripts** flow. Populate the current version and the target version input parameters. Set the target version parameter to 8.1. For example:

  CURRENT_TDM_VERSION = 8.0.

  TARGET_TDM_VERSION = 8.1.

6. Data Generation Implementation – PII Fields

- The implementation guidelines for the previous TDM version recommended to use the LU population's Masking Actors for the synthetic data generation as well. However, this approach was less effective when defining a PII field as an external input parameter for the data generation task.
- There are 2 alternatives for existing data generation flows:
  - Keep the existing implementation using the Masking actors. In order to enable the Masking Actors on data generation tasks, you must check the Mask Sensitive Data checkbox of the Synthetic environment window in the TDM portal.

  - Edit the data generation flows: use the random data generation Actors or the new TDM 8.1 Actor, GenerateConsistent.

    Click here for data generation's implementation guidelines.