TDM (TEST DATA MANAGEMENT) UPGRADE PROCEDURE TO V9.0

- This document describes the following:
  - How to upgrade TDM onto the present version: V9.0.
  - 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 9.0 Installation - Prerequisites

- Upgrade Fabric to Fabric 8.0 and above.

2. Related Documents

- FABRIC UPGRADE PROCEDURE TO V8.0
  - Note that Step 1 of the Fabric Upgrade Procedure document is irrelevant for a TDM project, since the TDM project does not contain the iidFinder process.
- For more information about TDM V9.0 installation, please read the TDM Installation article in the TDM Configuration.

3. TDM Upgrade

3.1 Upgrade the TDM Project and the TDM DB

3.1.1 Import the TDM 9 Library

3.1.1.1 Web Studio

Step 1 – Open the TDM Project in Fabric 8.0 Studio

- Open the TDM project with Fabric Studio 8.0. The Fabric Studio will upgrade the code to the latest version 8.0.

- Delete manually the following:
  - TDM LU
  - TDM_LIBRARY LU
  - TDM_Reference LU

- Click the Extension icon -  - and select TDM to install the TDM 9 library. Override the existing objects.

3.1.1.2 .Net (desktop) Studio

Step 1 – Open the TDM Project in Fabric 8.0 Studio

- Open the TDM project with Fabric Studio 8.0. The Fabric Studio will upgrade the code to the latest version 8.0.
- Copy the following .jar files into the \K2View Fabric Studio\Projects\<project name>\lib folder:
  - json-20231013
  - handlebars-4.3.0
  - cron-utils-9.2.1
  - commons-lang3-3.11

- Note that commons-lang3-3.11 is needed only for the upgrade flows and must be removed from the lib folder after the TDM upgrade since the TDM 9 code no longer uses the StringUtils object.
- Delete manually the following:
  - TDM LU
  - TDM_LIBRARY LU
  - TDM_Reference LU
- If the legacy project is based a Fabric version, which is older than 7.2, open the Environment window in the Studio, re-save it, and re-deploy the environments to Fabric debug server.

Step 2 – Import the TDM 9 Library into the Project

- Import the TDM LUs export file into your project using the 'Import All' option in order to import the following LUs:
  - TDM
  - TDM_LIBRARY LU
  - TDM_TableLevel LU

- Custom import the **Web Services** into your project.
- Custom import the following shared objects into the Fabric project:
  - **Templates**
  - **Broadway**
  - **Java**
- Custom import the **MTables** under the **Reference** to the project. If the legacy TDM version is 8.1 => remove the conflicting MTables from the import.
- **Optional – AI Interfaces**:
  - Import the AI interfaces if you wish to add the AI-based generation configuration to the TDM project. For more details, see the **AI Configuration** section below.

- Note that the import creates duplicated objects in the project, since TDM 9 locates the TDM objects in subfolders in the shared object's Broadway and Java folders. The duplicated objects will be removed by the next step (running the UpgradeTDMProjectToTDM9 flow).

Step 3 – Optional - Edit the Masking DB Global

- A new Global has been added in TDM 8.1 - SEQ_CACHE_INTERFACE. This Global is populated with the DB interface of the k2masking DB (PostgreSQL or Cassandra) and must be aligned with Fabric's system DB. TDM 9 sets the **POSTGRESQL_ADMIN** as a **default** value in this Global:
  - If you use **Cassandra** as Fabric's system DB, you must edit the SEQ_CACHE_INTERFACE Global and update its value to **DB_CASSANDRA**.
  - If you wish to use the **PostgreSQL** DB as Fabric system DB, do the following:
    - Open the Fabric's **config.ini** file and edit the **[system_db]** section's attributes including the SYSTEM_DB_DATABASE attribute to be aligned with the **POSTGRESQL_ADMIN** DB interface**.**
    - Restart Fabric.

3.1.2 Optional Steps

3.1.2.1 Rerun the Extract Tasks for Tables

- TDM 9 changes the way the tables are stored in Fabric. The tables are now stored in a new LU: **TDM_TableLevel.** The previous LU - **TDM_Reference** – is not longer in use.

3.1.2.2 AI Configuration

- Edit the AI interfaces –
  - If you do not have an AI installation for the AI-based generation – set the AI interfaces to be inactive.

- If you have an AI installation for the AI-based generation – add the AI environment and edit the AI interfaces inside it.

- Click here for more information about the AI implementation.

3.1.2.3 Update the Tasks with the Creator's Fabric Role

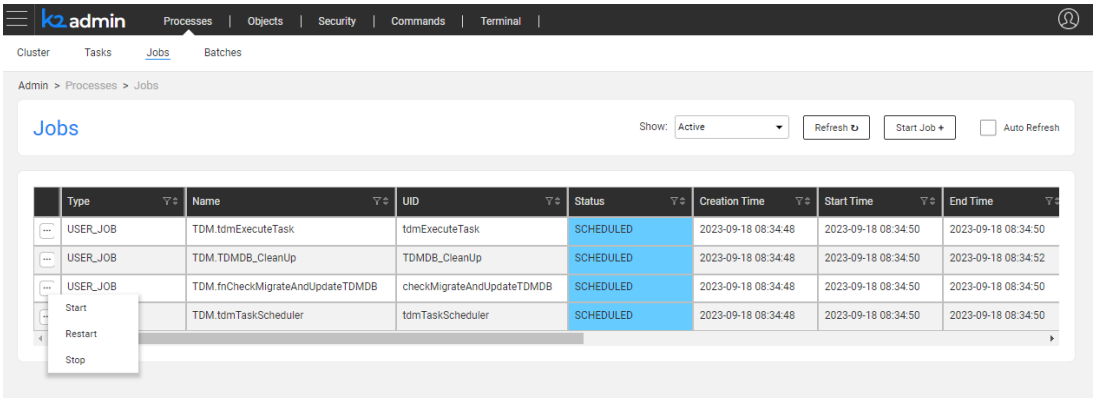
- This step is needed when the users are managed by an external IDP (e.g. SAML) and you use the TDM portal logic, which enables the following users to execute a task:
  - The task creator.
  - Additional users that belong to the task creator's group (Fabric role).
- From TDM 9.0 onwards, the user's Fabric role is concatenated to the user name in the Tasks TDM DB table. This is required in order to identify the task creator's Fabric role and decide if a tester user can execute a task, created by another user, when all users are managed and kept by the organization's IDP.
- Populate the **UserRolesUpgrade** MTable with the list of the TDM users and their Fabric roles before running the **RunTDMDBUpgradeScripts** flow. This flow will concatenate the user's Fabric role to the **task_created_by** and the **task_last_updated_by** fields of the Tasks TDM DB table.
- Redeploy the **Reference** LU.

3.1.2.4 Tables' Processing - Increase the Maximum Number of Records

- If you need to process tables (with or without entities) by the TDM tasks, note that the default limitation on number of processed records is 100K records. If your tables have a larger number of records, do the following:
  - Open the **config.ini** file and edit the **[broadway]** section – add the **MAX_CONCRETE_ARRAY_SIZE** attribute and set its value with a value larger than 100,000. For example:
    - `MAX_CONCRETE_ARRAY_SIZE=50000000`
  - Restart Fabric

3.1.3 Run the UpgradeTDMProjectToTDM9 flow

- Take the following steps **before** running the upgrade script:
  - Deploy the TDM LU. It is recommended to use the soft deploy option in order to avoid starting the TDM jobs.
  - Back up the TDM DB.
  - Deploy all LUs including the TDM LU to Fabric server.
  - Stop all TDM jobs on Fabric and execution servers in order to avoid locking the TDM DB tables by parallel executions of the upgrade script and the TDM jobs. Use the Web Admin to stop the TDM jobs:



- Open and run the **UpgradeTDMProjectToTDM9** flow in order to update the TDM project with the updated TDM library, convert the legacy TDM translations to MTables, and upgrade the TDM DB.
- Note:
    - The upgrade process retrieves the **legacy TDM version** from the **tdm_general_parameters** TDB DB table.

UPGRADE THE TDM DB IN THE EXECUTION SERVER

- If you do not have access to the TDM DB from the Studio debug server, do the following to upgrade the TDM DB:
    - Open the **Upgrade80_to_81** flow for edit and disable the **convertLuTranslations** Actor.
    - Open the **Upgrade81_to_90** flow for edit and disable the **UpdatePostProcessList** Actor.
    - Deploy the TDM LU to Fabric execution server.
    - Open a Fabric console in the Fabric execution server and run the **RunTDMDBUpgradeScripts** flow using the following command:

      Broadway TDM.RunTDMDBUpgradeScripts CURRENT_TDM_VERSION='<current version>', TARGET_TDM_VERSION='9.0';

    - Run the following flows in Fabric debug server, if needed:
        - **convertLuTranslations**
        - **UpdatePostProcessList**

3.1.4 Optional - Add the Catalog Masking Actor

- Open the LU populations and the data generation flows and add the CatalogMaskingMapper Actor to the flows if you wish to use the catalog masking. Get the value from the SEQ_CACHE_INTERFACE Global and send it to the interface parameter of the CatalogMaskingMapper.

3.2 Additional Steps – the Legacy TDM Version is Older than 8.1

3.2.1 TDM Portal – Re-saving Tasks with Parameters Selection Method

- TDM 8.1 changed the way the tasks with Parameters selection method are saved in the TDM DB. Therefore, it is required to open and resave TDM tasks with Parameters selection method after upgrading the TDM and before executing the TDM tasks.

3.2.2 Rerun an Extract Task to Repopulate the LUs Parameters' Tables

- The upgrade script **updates the <LU name>_params table and is based on the task_execution_entities**. By default, the task_execution_entities table contains executions only of the last 7 days (=0.25 month). Therefore, **the <LU name>_params table will also contain only the entities of the last 7 days executions** (if the related task_execution_entities record is not found, the upgrade job will delete the related <LU name>_params record as well).
- If the <LU name>_param table must contain executions history longer than the last 7 days, rerun an Extract task on a large population, after the TDM upgrade, in order to repopulate the missing <LU name>_params records.

3.2.3 Manual Updates

- Open **TDMFilterOutTargetTables** Actor and add the following Boolean column if it is missing: **generator_filterout**. Set it to **true** for all the TDM product tables and the _TAR table, setting it to true.
- Open **CustomLogicFlows** Actor and add the following Boolean column if it is missing: **DIRECT_FLOW**. Leave this field cleared for the existing records.

  Click here for more information about Custom Logic implementation.

- Note that the TDM translations will be converted to MTables by the next step – running the **RunTDMDBUpgradeScripts** flow.

3.3 Deployment

- Deploy all the LUs in the project including the Reference and Web Services LUs.
- Verify that the TDM jobs are up and running.

4. Optional - Change Fabric Storage to a Storage that does not Support a TTL

- TDM enables creating tasks with a retention period (TTL) on the task's entities in order to save these entities in Fabric only for a limited period of time. However, if the Fabric storage does not support TTL for the LUIs (such as PG DB), the TDM needs to limit the TDM task's retention period options to either 'Do not Delete' or 'Do not Retain'.
- Run the following steps to limit the TDM retention period:
  I. Update the tdm_general_parameters TDM DB to limit the TDM task's retention period options to either 'Do not Delete' or 'Do not Retain'.

  View the Update statements in
  https://support.k2view.com/Academy/articles/TDM/tdm_configuration/02_tdmdb_general_parameters.html

  II. Open the TDM portal, then open the TDM tasks and update them with a retention period other than 'Do not Delete' or 'Do not Retain'.