TDM (TEST DATA MANAGEMENT) UPGRADE PROCEDURE TO V9.1

- This document describes the following:
  - How to upgrade TDM from TDM 9.0 onto the present version - V9.1.
  - 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 9.1 Installation - Prerequisites

- Upgrade Fabric to Fabric V8.1 and above.

2. Related Documents

- FABRIC UPGRADE PROCEDURE TO V8.1
  - 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 V9.1 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 9.1 Library

3.1.1.1 Web Studio

Step 1 – Open the TDM Project in Fabric 8.1 Studio

- Open the TDM project with Fabric Studio 8.1. The Fabric Studio will upgrade the code to the latest version, 8.1.

- 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 – Back up the Project's Populated TDM Objects

- Back up the following objects in your project:
  - CustomLogicFlows.actor
  - TDMFilterOutTargetTables.actor
  - TDMSeqList.actor
  - TDMSeqSrc2TrgMapping.actor
  - TDMTargetTablesNames.actor
  - TableLevelInterfaces.csv

Step 2 – Open the TDM Project in Fabric 8.1 Studio

- Open the TDM project with Fabric Studio 8.1. The Fabric Studio will upgrade the code to the latest version, 8.1.
- Manually delete the following:
  - TDM LU
  - TDM_LIBRARY LU

Step 3 – Import the TDM 9.1 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:

- TableLevelDefinitions.csv
- Note that the TableLevelInterfaces.csv will be updated by the TDM deploy flow.

- **Optional – AI Interfaces**:
  - Import the AI interfaces if you wish to add the AI-based generation configuration to the TDM project.
  - 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 Upgrade the TDM DB

- Deploy the updated TDM LU. The TDM deploy flow runs the **TDMDBUpgradeScripts** flow in order to upgrade the TDM DB.

4. Optional – Change the Parameters mode to Parameters Coupling

- TDM 9.1 introduces a new mode when selecting an entity subset for a TDM task based on business parameters: *Parameters Coupling*.
- The following steps are needed if you wish to set the parameter's mode to parameters coupling:

a. Set, Create and Alter Schema and Table Permissions for the TDM User

- The parameters coupling mode uses the MDB export Fabric command in order to export the parameters' info of each LU into a dedicated schema in the TDM DB. A separate schema is created in the TDM DB for each LU.
- Verify that the TDM DB user, which is set in the TDM interface, has permissions to create and edit schemas and tables.

b. Run the UpgradeToParamsCouplingMode Flow

- The UpgradeToParamsCouplingMode flow executes the following:
  - Updates the TDM_GENERAL_PARAMETERS TDM DB table – sets the PARAM_COUPLING parameter to 'true'.
  - Creates a backup table for the tdm_params_distinct_values TDM DB table.
  - Truncates the tdm_params_distinct_values in the TDM DB table.
  - Renames the <lu>_params tables in the TDM DB – adds a _bck suffix to these tables since they are no longer needed for the parameters coupling mode.
  - Converts the LuParams.csv to the LuParamsMapping.csv if possible.
  - Adds the TDM_BE_IIDS LU table to the LUs.

- Updates the FABRIC_TDM_ROOT table in the LU – adds PK to this table in order to support the MDB export of the LU into the TDM DB.

c. Update the LU Implementation

- Verify that the linked fields in the LU tables have identical data types. This is required in order to support the MDB export of the LU schema into the TDM DB.
- Add the TDM_BE_IIDS table to **TDMFilterOutTargetTables.actor** in order to exclude these tables from the load, delete, and data generation flows.
- Add a table to the LU for calculated parameters. For example, the total open debt amount is based on the accumulation of all open invoices. Each parameter in the parameters coupling mode must be mapped to an LU table's field. Unlike the regular mode, you cannot define an SQL query to get a parameter in the LuParamsMapping Mtable.
- Add the new tables with the calculated parameters to **TDMFilterOutTargetTables.actor** in order to exclude these tables from the load, delete, and data generation flows.

  Verify that all the LU tables in the LuParamsMapping are linked to parent tables. This is required in order to add an FK to tables when they are exported to the TDM DB.
- Update the LuParamsMapping.csv MTable – add the parameters that are based on the newly created business tables.
- Redeploy the implementation, including the Reference LU.

d. Rerun the Extract Tasks

- Re-extract an entity subset for each Business Entity (BE) in order to:
  - Create the LU schemas in the TDM DB and export the entities' data into these schemas.
  - Re-populate the tdm_params_distinct_values in the TDM DB table.